Superseded by a later version of this document.

OpenCable™ Specifications
ETV

Enhanced TV Application Messaging Protocol 1.0

OC-SP-ETV-AM1.0-I05-091125

ISSUED

Notice

This OpenCable specification is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. This document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in the document.

Neither CableLabs nor any member company is responsible to any party for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document, or any document referenced herein. This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, or fitness for a particular purpose of this document, or any document referenced herein.

© Copyright 2004-2009 Cable Television Laboratories, Inc.
All rights reserved.

Charter Ex. 1153

Document Status Sheet

Document Control Number:	OC-SP-ETV-AM1.0-I05-091125			
Document Title:	Enhanced TV Application Messaging Protocol 1.0			
Revision History:	I01 - Issued April 18, 2005			
	I02 - Issued July 27, 2005			
	I03 - Issued July 14, 2006			
	I04 - Issued September 21, 2007			
	I05 - Issued November 25, 2009			
Date:	November 25, 2009			
Status:	~~Work in Progress~~	~~Draft~~	Issued	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	Public

Key to Document Status Codes:

Work in Progress	An incomplete document, designed to guide discussion and generate feedback, that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks:

CableLabs®, DOCSIS®, EuroDOCSIS™, eDOCSIS™, M-CMTS™, PacketCable™, EuroPacketCable™, PCMM™, CableHome®, CableOffice™, OpenCable™, OCAP™, CableCARD™, M-Card™, DCAS™, tru2way™, and CablePC™ are trademarks of Cable Television Laboratories, Inc.

Contents

1 SCOPE 1

1.1 INTRODUCTION AND OVERVIEW 1
1.2 PURPOSE OF DOCUMENT 1
1.3 ORGANIZATION OF DOCUMENT 1
1.4 REQUIREMENTS 2

2 REFERENCES 3

2.1 NORMATIVE REFERENCES 3
2.2 INFORMATIVE REFERENCES 3
2.3 REFERENCE ACQUISITION 3

3 TERMS AND DEFINITIONS 4

4 ABBREVIATIONS AND ACRONYMS 5

5 ENHANCED TELEVISION APPLICATION MODEL 6

5.1 ETV APPLICATIONS 6
5.2 ETV AUTHORING PROCESS 6
5.2.1 Production Team 6
5.2.2 Network Broadcaster 6
5.2.3 Cable Operators 7
5.2.4 Application Servers 7
5.3 ENHANCED TELEVISION COMPONENTS 7
5.4 MEDIA TIMELINE 8

6 ENHANCED TELEVISION APPLICATION SIGNALING 10

6.1 INTRODUCTION 10
6.2 PROGRAM MAP TABLE DESCRIPTORS 10
6.2.1 ETV Registration Descriptor 10
6.2.2 ETV Integrated Signaling Descriptor 11
6.2.3 ETV-BIF Platform Descriptor 11
6.3 APPLICATION SIGNALING FOR ANALOG SERVICES 13

7 ENHANCED TELEVISION SYNCHRONIZATION SIGNALING 14

7.1 EISS TABLE 14
7.2 EISS DESCRIPTORS 16
7.2.1 ETV Application Information Descriptor 16
7.2.2 ETV Media Time Descriptor 19
7.2.3 ETV Stream Event Descriptor 19
7.3 SYNCHRONIZATION IN ANALOG SERVICES 20

8 CARRIAGE OF ETV APPLICATION RESOURCE DATA 21

8.1 DSM-CC DATA CAROUSEL 21
8.2 ALTERNATE CONSTRAINED DATA CAROUSELS 22

9 APPLICATION SIGNALING AND SYNCHRONIZATION FOR LIMITED CAPABILITY DEVICES 24

9.1 INTRODUCTION 24
9.2 ALL OTHER SET-TOP SPECIFIC BEHAVIORS 25
9.3 OPENCABLE HOST SPECIFIC BEHAVIORS 25

APPENDIX I REVISION HISTORY 26

List of Figures

FIGURE 1 - ENHANCEMENT DISTRIBUTION PROCESS 7
FIGURE 2 - ENHANCED TELEVISION COMPONENTS 8
FIGURE 3 - PMT SIGNALING MOTOROLA DCT-2000 SPECIFIC BEHAVIORS 24

List of Tables

TABLE 1 - ETV REGISTRATION DESCRIPTOR SYNTAX 10
TABLE 2 - ETV INTEGRATED SIGNALING DESCRIPTOR SYNTAX 11
TABLE 3 - ETV-BIF PLATFORM DESCRIPTOR SYNTAX 12
TABLE 4 - ETV-BIF PLATFORM ID SYNTAX 12
TABLE 5 - EISS SECTION SYNTAX 15
TABLE 6 - APPLICATION TYPES 16
TABLE 7 - ETV APPLICATION INFORMATION DESCRIPTOR SYNTAX 17
TABLE 8 - ETV-BIF APPLICATION CONTROL CODE VALUES 17
TABLE 9 - ETV-BIF APPLICATION VERSION 17
TABLE 10 - ETV-BIF APPLICATION FLAGS 18
TABLE 11 - ETV MEDIA TIME DESCRIPTOR SYNTAX 19
TABLE 12 - ETV STREAM EVENT DESCRIPTOR SYNTAX 20
TABLE 13 - ABS_PATH 21
TABLE 14 - AUTHORITY 22
TABLE 15 - DCII DATA CAROUSEL MESSAGE SYNTAX 22

1 SCOPE

1.1 Introduction and Overview

Broadcasters and network operators around the world are deploying interactive applications by creating enhancements to a broadcast video stream. These Enhanced Television (ETV) applications rely on embedding various types of data in the video stream, including programs, images, and triggers.

This document specifies the synchronization and signaling mechanisms to be used by ETV applications, regardless of the target receiver or middleware environment. ETV mechanisms must be implementable by legacy set-top boxes as well as OpenCable (OCAP) host devices, and this implementation requirement implies that more than one option must exist for the physical transmission of the signaling and trigger data. This document addresses those various options and describes how a set-top box should interpret signals and triggers delivered via each of those methods.

1.2 Purpose of document

The purpose of this document is to specify ETV application signaling and synchronization mechanisms that meet all of the objectives/requirements of North American cable systems for delivering video-synchronous ETV applications, whether they are broadcast or delivered on-demand.

The intent is to propose a uniform method of inserting signals and triggers that is independent of application environments and software/technology vendors. That said, it is understood that accommodations must be made for the support of specific legacy set-top boxes such as the DCT-2000 and Explorer 2000, while also supplying a standard rich enough to work with advanced set-top boxes based on the OpenCable Host 2.0 Core Functional Requirements [HOST2.1].

In some cases, the need to support a range of devices may result in the need to have more than one signaling packet delivered through the network for the same application. As the number of legacy set-top boxes drops to zero, in any given division over the next several years, this requirement would be relaxed.

This document does not attempt to impose a selection of a particular vendor for implementation. The design of the system is largely based on open industry standards with an objective to leverage currently existing equipment and tools available for implementing such a system.

1.3 Organization of document

This document is divided into four parts:

- a description of the type of applications to be addressed by this specification,
- application signaling and life-cycle management,
- application synchronization and timeline management,
- platform-specific constraints imposed by legacy environments.

1.4 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"SHALL"	This word means that the item is an absolute requirement of this specification.
"SHALL NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works (or portions thereof as indicated in this specification), in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[IEC 13818-1] ISO/IEC 13818-1:2000, Information technology—Generic coding of moving pictures and associated audio information: Systems, 2000.

[IEC 13818-6] ISO/IEC 13818-6:1998(E), Information technology—Generic coding of moving pictures and associated audio information: Extensions for DSM-CC, 1998.

[OCAP1.1] OC-SP-OCAP1.1.2-090930, OpenCable Application Platform Specification, Profile 1.1, September 30, 2009, Cable Television Laboratories, Inc.

[MHP] ETSI TS 101 812 V1.3.1 (2003-06), DVB Multimedia Home Platform (MHP) 1.0.3.

[ETV-BIF] OC-SP-ETV-BIF1.0-I05-091125, OpenCable Enhanced Television (ETV) Binary Interchange Format 1.0, November 25, 2009, Cable Television Laboratories, Inc.

[SMPTE343M] SMPTE 343M-2002, Declarative Data Essence – Local Identifier (lid:) URI Scheme.

2.2 Informative References

[HOST2.1] OC-SP-HOST2.1-CFR-I09-090904, OpenCable Host 2.1 Core Functional Requirements, September 4, 2009, Cable Television Laboratories, Inc.

2.3 Reference Acquisition

CableLabs Specifications:

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027;
  Phone 303-661-9100; Fax 303-661-9199; Internet: http://www.cablelabs.com

ETSI Specifications:

- European Telecommunications Standards Institute, http://www.etsi.org

ISO Specifications:

- International Organization for Standardization, http://www.iso.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Application Signal	A broadcast message that provides information to a receiver necessary to acquire, launch, and terminate an ETV application.
Enhanced Television (ETV)	A general term that refers to interactive services and applications provided in conjunction with video programming.
Enhancement	A software application that executes in conjunction with video programming.
Trigger	A broadcast message that provides a synchronization mechanism to an enhancement. Triggers may be embedded in the associated video program, or delivered via another means such as OOB. Triggers may also be used for the delivery of unsolicited data to an enhancement. Triggers may include application signals and stream events.
Stream Event	A type of Trigger that conveys application defined messages to an enhancement. ETV stream events are normatively defined within this specification.
User Agent	An application running on a receiver that decodes and executes the enhancement.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

AIT	Application Information Table
bslbf	bit-string left bit first
DCII	DigiCipher II
DSM-CC	Digital Storage Module - Command and Control
DTD	Document Type Definition
DVR	Digital Video Recorder
EBIF	ETV Binary Interchange Format
ETV	Enhanced Television
EISS	ETV Integrated Signaling Stream
IB	In-band
NPT	Normal Play Time
OOB	Out-of-Band
PMT	Program Map Table
PTS	Presentation Time Stamp
rpchof	remainder polynomial coefficients, highest order first
uimsbf	unsigned integer most significant bit first
URI	Uniform Resource Identifier
VBI	Vertical Blanking Interval

5 ENHANCED TELEVISION APPLICATION MODEL

5.1 ETV Applications

This specification is intended to support a wide variety of program synchronous applications, such as:

- Interactive Advertising
- Game Shows
- News
- Sports Events
- Voting applications
- Impulse upgrade promotions
- E-commerce applications

This specification addresses both live broadcasts and pre-recorded programs and supports “real-time” viewing as well as time-shifted (DVR) viewing and interaction.

5.2 ETV Authoring Process

There are several important factors in the creation and deployment of ETV applications:

- Production Team
- Cable Operator
- Network Broadcaster
- Application Server

Figure 1 provides a graphical illustration of the relationships between these functional groups.

5.2.1 Production Team

Application production teams generate the interactive enhancements in conjunction with the studios that produce the video. Applications are often built around templates for the triggers and data that are inserted by a production team.

For pre-recorded shows, the application signaling and triggers are mastered during the video post-production process before the show is broadcast. Throughout the production process, the production team uses a media timeline based on SMPTE time codes, which are used for synchronization of video tape.

For live shows, the application itself may be prepared in advance, but the actual data to feed the application is inserted dynamically—for example: Who just won the Oscar for Best Director?

5.2.2 Network Broadcaster

Enhancements cannot be bound to a program until after the video has been digitized for final broadcast. Furthermore, the video feeds from the major networks are often converted back to analog by a local network affiliate and then re-digitized for use by the local cable company, losing many enhancements along the way. Cable channels have the advantage of staying in the digital domain and so can insert all enhancements at the digital encoding stage.

Applications and triggers must be inserted into the video stream by the broadcaster using synchronization triggers and play lists that are appropriate for the type of equipment used in the broadcast environment. At the same time, broadcasters may have to rely on sending those enhancements across a broadband connection to cable companies and affiliates who do not preserve the full digital signal from beginning to end. Some enhancements can be sent

ahead of time for automatic insertion on a given schedule. Others may be transmitted in real time and synchronized to live events—either by production teams directly, or via the broadcasters or the cable operators.

5.2.3 Cable Operators

Some cable operators generate their own programming and enhancements. Again, they are responsible for obtaining the enhancement from the Production Team and inserting the application, signaling and triggers into the video streams, adding to or replacing video segments received from the satellite. At this point, cable operators may be working in either the analog or digital domain, because the video may be in either state before final transmission to the subscriber. Unfortunately, analog and digital video place different requirements on the data insertion equipment.

5.2.4 Application Servers

Although it probably does not signal a base enhancement directly, a polling or score server MAY be used to dynamically process subscriber votes or quiz answers and send responses back to particular client receivers. Messages sent from the polling server MAY require routing or insertion by the cable operator back into either an IB or OOB data stream to a client.

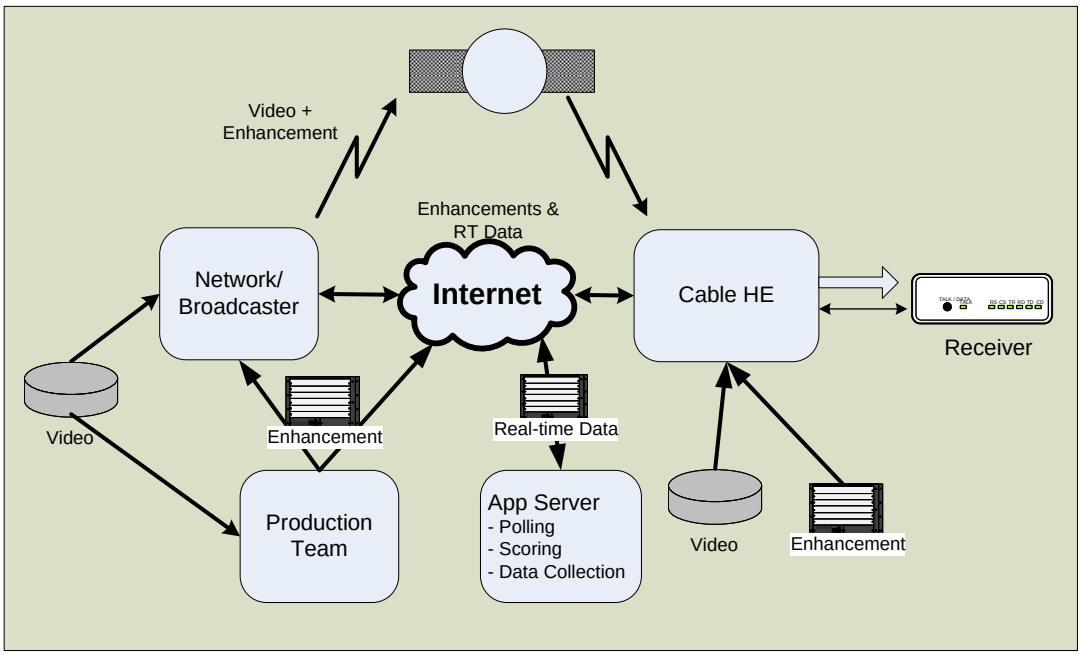

Figure 1 - Enhancement Distribution Process

5.3 Enhanced Television Components

An enhanced television application is made up of several components:

- Broadcast Audio/Visual program
- ETV Application Program and Resource Data
- Application Signaling
- Stream Events/Triggers
- Media Timeline

In addition, an ETV application MAY send data to an application server. Responses from the server MAY come in the form of new application resources or stream events.

These components are illustrated in Figure 2.

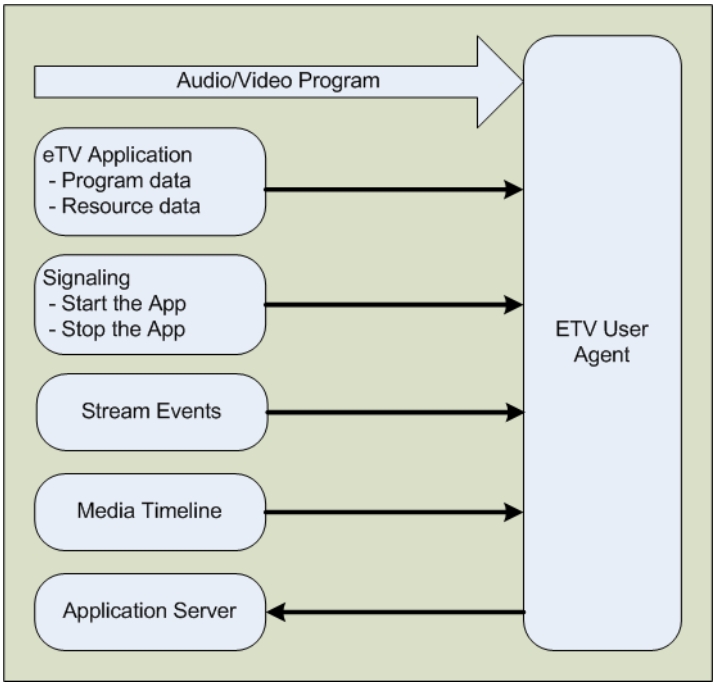

Figure 2 **- Enhanced Television components**

While all of these components together complete the ETV application, it helps to look at them separately in the context of discussing their delivery and formats:

- The **ETV application** is the subject of other specifications, such as OpenCable Enhanced Television (ETV) [ETV-BIF], although this specification will discuss the delivery and encapsulation of the ETV application data.
- **Signaling** commands defined in this specification tell the ETV User Agent in the client receiver to start and stop the application, and identify how to find and load the primary elements of the application.
- **Stream Events** deliver "unsolicited" data to an application, synchronized with the video stream.
- The **Media Timeline** provides the basic timing references to which stream events are synchronized.
- ETV **Application Servers** are used for collecting the results of an ETV application, such as requests for additional information or storing poll or voting responses.

One advantage of separating the ETV application format from this signaling specification is that this specification can be used to signal a number of different types of applications, including ones written to OpenCable Enhanced Television (ETV) [ETV-BIF], as well as to other specifications yet to be written. In general, the problems of signaling and media synchronization are universal, regardless of the type of application being signaled.

5.4 Media Timeline

A Media Timeline is a continuous timeline over the duration of an event. An event is defined in ISO/IEC 13818.1 as a collection of elementary streams with a common time base, an associated start time, and an associated end time. A typical, but not normative example, is the video and audio for a single television show.

The Media Timeline refers to the real time of the event. For example, when an event is presented in reverse, the timeline counts down rather than up; and when an event is presented at 10 times the normal rate, the timeline progresses at 10 times the normal rate. In this way, the Media Timeline increases and decreases in a way similar to a counter on a video tape recorder and provides an absolute timeline to which references can be made for operations such as jumping to a particular point in the event.

Media timelines can be paused, for example, during a commercial break which occurs interstitially between two segments of the event. Media timelines can also be nested, for example, if the commercial itself contains an enhancement.

6 ENHANCED TELEVISION APPLICATION SIGNALING

6.1 Introduction

This section defines the data formats and transmission mechanisms that enable receivers to discover, download, launch, and terminate ETV applications.

6.2 Program Map Table Descriptors

The following PMT descriptors are defined by this specification:

- ETV registration descriptor
- ETV integrated signaling descriptor
- ETV-BIF platform descriptor

6.2.1 ETV Registration Descriptor

The `etv_registration_descriptor` is defined for use in the elementary stream information loop of the PMT. The ETV registration descriptor [IEC 13818-1], table 2-46 – Registration Descriptor, clause 2.6.8) is defined to identify unambiguously the programs that comply with this specification. The ETV registration descriptor SHALL be carried in the elementary stream information loop of the PMT for each program component that conveys an ETV integrated signaling stream or ETV application resource stream. The presence of the registration descriptor uniquely and unambiguously identifies the formats of the private data used within this specification, including private use fields and identifiers within *user private* ranges such as descriptor tags and MPEG private section table ids.

The `etv_registration_descriptor` is defined in Table 1.

Table 1 - ETV Registration Descriptor Syntax

Syntax	Bits	Mnemonic
`etv_registration_descriptor() {`		
`  descriptor_tag`	8	uimsbf
`  descriptor_length`	8	uimsbf
`  etv_format_identifier`	32	bslbf
`}`		

descriptor_tag This 8-bit integer with value `0x05` identifies this descriptor.

descriptor_length This 8-bit integer indicates the number of bytes following the descriptor length field.

etv_format_identifier CableLabs has assigned a value of 0x45545631 (ASCII “ETV1”) to this 4-byte field to identify the program component (within a multiplex) in which it is carried as complying with this specification.

Note: This value must be registered with the SMPTE Registration Authority, LLC and is subject to change. SMPTE is authorized by the International Organization for Standardization (ISO) to register MPEG-2 format identifiers per [IEC 13818-1]:2000 Annex O: Registration procedure [for "format_identifier"].

6.2.2 ETV Integrated Signaling Descriptor

The `etv_integrated_signaling_descriptor` is defined for use in the elementary stream information loop of the PMT. This descriptor indicates that the associated elementary stream contains an ETV integrated signaling stream, as defined in Section 7 of this specification. Only one elementary stream signaled by the PMT SHALL contain an `etv_integrated_signaling_descriptor`. In the event that more than one PMT entry contains an `etv_integrated_signaling_descriptor`, the behavior of the ETV receiver is undefined.

A PMT entry with an `etv_integrated_signaling_descriptor` MAY be associated with a stream_type of 0xC0 or 0x05.

The `etv_integrated_signaling_descriptor` is defined in Table 2.

Table 2 - ETV Integrated Signaling Descriptor Syntax

Syntax	Bits	Mnemonic
`etv_integrated_signaling_descriptor() {`		
`  descriptor_tag`	8	uimsbf
`  descriptor_length`	8	uimsbf
`  platform_id_length`	8	uimsbf
`  for (i=0; i<platform_id_length; i++){`		
`  etv_bif_platform_ids()`		
`  }`		
`  for (i=0; i<n; i++) {`		
`  private_use`	8	bslbf
`  }`		
`}`		

descriptor_tag This 8-bit integer with value `0xA2` identifies this descriptor.

descriptor_length This 8-bit integer indicates the number of bytes following the descriptor length field.

platform_id_length This 8-bit integer specifies the number of bytes of the `etv_bif_platform_ids` field immediately following this `platform_id_length` field. This value MAY be zero.

etv_bif_platform_ids This field contains zero or more platforms, each identified by an `etv_bif_platform_id` as defined in Table 4. If the associated elementary stream is NOT intended to be decoded on all platforms, this descriptor SHALL list all supported `etv_bif_platform_ids`.

If the `platform_id_length` is zero (i.e., there are no etv_bif_platform_ids listed by this descriptor), the associated elementary stream SHALL be decoded on ALL platforms.

private_use This field may be used to carry private data to a receiver or user agent which interprets this signaling stream. Its use is not defined by this specification.

6.2.3 ETV-BIF Platform Descriptor

The `etv_bif_platform_descriptor` is defined for use in the elementary stream information loop of the PMT. This descriptor indicates to a receiver that the associated elementary stream carries ETV-BIF application resources for one or more hardware/software platforms (as identified by [ETV-BIF]) each of which SHALL be listed in this descriptor.

Resources that target baseline receivers SHALL be transmitted on a single elementary stream. Resources that target other receivers MAY be transmitted across multiple elementary streams. In this case, an `etv_bif_platform_descriptor`, which identifies the targeted platform, SHALL be included in the PMT entry that corresponds to each elementary stream carrying those resources.

An elementary stream MAY contain resources that target different collections of platforms. An elementary stream MAY also contain resources for multiple applications; (for example, resources for the primary broadcast program and resources for an enhanced advertisement). The PMT elementary stream information loop MAY contain multiple instances of an `etv_bif_platform_descriptor`.

The `etv_bif_platform_descriptor` is defined in Table 3.

Table 3 - ETV-BIF Platform Descriptor Syntax

Syntax	Bits	Mnemonic
`etv_bif_platform_descriptor() {`		
`  descriptor_tag`	8	uimsbf
`  descriptor_length`	8	uimsbf
`  for (i=0; i < n; i++) {`		
`  etv_bif_platform_ids()`		
`  }`		
`}`		

descriptor_tag This 8-bit integer with value `0xA1` identifies this descriptor.

descriptor_length This 8-bit integer indicates the number of bytes following this field.

etv_bif_platform_id This field contains zero or more platforms, each identified by an etv_bif_platform_id as defined in Table 4. If the associated elementary stream contains resources for more than one platform, this descriptor SHALL list all supported `etv_bif_platform_ids`.

If descriptor_length is zero (i.e., there are no etv_bif_platform_ids listed by this descriptor), the associated elementary stream SHALL carry resources that target ALL platforms.

Table 4 - ETV-BIF Platform ID Syntax

Syntax	Bits	Mnemonic
`etv_bif_platform_id() {`		
`  pdtHWManufacturer`	24	uimsbf
`  pdtHWModel`	16	uimsbf
`  pdtHWVersionMajor`	8	uimsbf
`  pdtHWVersionMinor`	8	uimsbf
`  pdtSWManufacturer`	24	uimsbf
`  pdtSWModel`	16	uimsbf
`  pdtSWVersionMajor`	8	uimsbf
`  pdtSWVersionMinor`	8	uimsbf
`  pdtProfile`	8	uimsbf
`}`		

All fields in this table are defined in [ETV-BIF], section 9.5.

6.3 Application Signaling for Analog Services

Application Signaling for analog services is out-of-scope for this specification.

7 ENHANCED TELEVISION SYNCHRONIZATION SIGNALING

Synchronization of an application to a video program requires the establishment of a reference media timeline. A media timeline allows a receiver to deliver stream events to an application at very specific points within the timeline.

OCAP devices are capable of using NPT descriptors and stream events embedded within a DSM-CC Object Carousel for this purpose as defined in sections 8.1 and 8.3 of DSM-CC [IEC 13818-6]. NPT descriptors establish the reference timeline. Stream events carry synchronous application data. Limited-capability devices, however, are not capable of properly interpreting and processing the DSM-CC Object Carousel and the descriptors carried therein.

This section defines a set of descriptors which all devices are capable of receiving, designed for maintaining a media timeline and delivering synchronous data to an application.

Because of the limitations of the very low-end legacy receivers, there are requirements to combine the media timeline messages and stream events into a single, unified elementary stream. This unified stream SHALL also carry additional descriptors as identified in this section.

This unified stream SHALL be known as an ETV integrated signaling stream (EISS). It is possible that an operator MAY deliver an EISS as well as a DSM-CC object carousel with embedded NPT & stream event descriptors. In this case, the receiver MAY use the media timeline that is best suited for its capabilities.

Only one EISS stream SHALL be conveyed for an MPEG-2 program. The EISS stream may carry EISS tables for multiple applications, each identified by an application identifier. An individual EISS table for an application SHALL NOT be interleaved with EISS table sections of another application. Therefore, this specification recommends that individual EISS tables be kept small to avoid any latency in delivering the EISS tables for other applications.

7.1 EISS Table

The descriptors defined in this section are carried in an EISS Table. This table is contained in one or more MPEG-2 sections with syntax as specified in Table 5.

Table 5 - EISS Section Syntax

Syntax	Bits	Mnemonic
`eiss_section() {`		
`  table_id`	8	uimsbf
`  section_syntax_indicator`	1	bslbf
`  reserved1`	3	bslbf
`  section_length`	12	uimsbf
`  reserved2`	8	uimsbf
`  section_number`	8	uimsbf
`  last_section_number`	8	uimsbf
`  protocol_version_major`	8	uimsbf
`  protocol_version_minor`	8	uimsbf
`  application_type`	16	uimsbf
`  application_identifier()`	48	
`  platform_id_length`	8	uimsbf
`  for (i=0; i<platform_id_length; i++) {`		
`  etv_bif_platform_ids()`		
`  }`		
`  for (i=0; i<N; i++) {`		
`  eiss_descriptor()`		
`  }`		
`  CRC_32`	32	rpchof
`}`		

table_id This 8-bit integer with value `0xE0` identifies this table.

section_syntax_indicator The `section_syntax_indicator` is a 1-bit field that SHALL be set to `0`.

reserved1 This 3-bit field SHALL be set to `000`.

section_length This is a 12-bit field that specifies the number of bytes of the section starting immediately following the `section_length` field, up to and including the `CRC_32` field. The value in this field SHALL not exceed 1021.

reserved2 This 8-bit field SHALL be set to `0x00`.

section_number This 8-bit field gives the number of the section. The `section_number` of the first section in the `sub_table` SHALL be `0x00`. The `section_number` SHALL be incremented by 1 with each additional section with the same `table_id`.

last_section_number This 8-bit field specifies the number of the last section (that is, the section with the highest `section_number`) of the sub_table of which this section is part.

protocol_version_major This 8-bit field specifies a major version number for the ETV messaging protocol being conveyed in the ETV streams for the currently signaled application. The required value is expected to be incremented each time a backward-incompatible revision is published to a fundamental syntactic structure of the ETV-AM protocol. The major version number corresponding to this specification is 3.

If the specified major version for the ETV messaging protocol is not supported and the application has not yet been loaded, then the user agent SHALL ignore the application being signaled; otherwise, if the application has been loaded, then the user agent SHALL cause an `UnsupportedVersionError` condition to be signaled, and subsequently, a terminating transition SHALL be triggered.

protocol_version_minor This 8-bit field specifies a minor version number for the ETV messaging protocol being conveyed in the ETV streams for the currently signaled application. The required value is expected to be incremented each time a backward-compatible revision is published to a fundamental syntactic structure of the ETV-AM protocol. The minor version number corresponding to this specification is 0.

If the specified minor version for the ETV messaging protocol is not supported, then the user agent SHALL cause an `UnsupportedVersionError` condition to be signaled once a loading transition has been effected.

application_type This 16-bit integer identifies the type of application being signaled. The following application types are defined:

Table 6 - Application Types

application_type	Description
0x0008	ETV-Binary Interchange Format (ETV-BIF) application

application_identifier This 48-bit integer identifies the application according to DVB-MHP section 10.5 [MHP].

platform_id_length This 8-bit integer specifies the number of bytes of the `etv_bif_platform_ids` field immediately following this `platform_id_length` field. This value MAY be zero.

etv_bif_platform_ids This field contains zero or more platforms, each identified by an `etv_bif_platform_id` as defined in Table 4. If zero, this eiss_section applies to all platforms. If non-zero, this field identifies the set of platforms for which this eiss_section applies. All of the eiss_sections for a single eiss table SHALL specify the same set of platforms.

eiss_descriptor Zero or more descriptors as specified in Section 7.2.

CRC_32 This 32-bit field SHALL be set as defined in [IEC 13818-1], Annex B.

7.2 EISS Descriptors

The contents of this section are applicable when the value of the application_type field of the eiss_section equals 0x0008 (ETV_BIF). This section defines the following EISS descriptors:

- ETV Application Information Descriptor
- ETV Media Time Descriptor
- ETV Stream Event Descriptor

7.2.1 ETV Application Information Descriptor

Because limited-capability devices cannot process normal AITs, the relevant fields from the AIT SHALL be embedded in the EISS as an application information descriptor, described in Table 7.

Table 7 - ETV Application Information Descriptor Syntax

Syntax	Bits	Mnemonic
`etv_application_information_descriptor() {`		
`  descriptor_tag`	8	uimsbf
`  descriptor_length`	8	uimsbf
`  application_control_code`	8	uimsbf
`  application_version()`		
`  max_protocol_version_major`	8	uimsbf
`  max_protocol_version_minor`	8	uimsbf
`  application_flags()`		
`  application_priority`	8	uimsbf
`  initial_resource_locator()`		
`  for (i=0; i<n; i++) {`		
`  private_data`	8	bslbf
`  }`		
`}`		

descriptor_tag This 8-bit integer with value `0xE0` identifies this descriptor.

descriptor_length This 8-bit integer indicates the number of bytes following this field.

application_control_code This 8-bit integer controls the state of the application. The semantics of this field are application type-dependent. This field is interpreted according to Table 8.

Table 8 - ETV-BIF Application Control Code Values

Code	Identifier	Semantics
0x00		reserved_for_future_use
0x01	AUTOSTART	The primary application resource is loaded and the application is started, subject to the usual restrictions, etc.
0x02	PRESENT	The primary application resource is loaded, but is not started, pending the receipt of another trigger, or Application Information descriptor.
0x03	DESTROY	The application is signaled to quit by generating a DESTROY event to the application.
0x04-0xff		reserved_for_future_use

application_version This 16-bit field carries the application version. It is structured according to Table 9.

Table 9 - ETV-BIF Application Version

Syntax	Bits	Mnemonic
`application_version () {`		
`  version_major`	8	uimsbf
`  version_minor`	8	uimsbf
`}`		

version_major This 8 integer carries the major version of the application.

version_minor This 8 integer carries the minor version of the application.

max_protocol_version_major This 8-bit field, if non-zero, specifies the maximum major protocol version supported by user agents that should decode this application. An ETV user agent that supports EISS protocol versions greater than max_protocol_version_major SHALL ignore this descriptor. This enables the user agent to ignore prior app version signaling in cases where a later version of the app is being signaled in this or subsequent EISS sub-tables.

max_protocol_version_minor This 8-bit field, if non-zero, specifies the maximum minor protocol version supported by user agents that should decode this application. An ETV user agent that supports EISS protocol versions greater than max_protocol_version_minor SHALL ignore this descriptor. This enables the user agent to ignore prior app version signaling in cases where a later version of the app is being signaled in this or subsequent EISS sub-tables.

application_flags This 32-bit field carries application flags. It is structured according to Table 10.

Table 10 - ETV-BIF Application Flags

Syntax	Bits	Mnemonic
`application_flags() {`		
`  test_flag`	8	bslbf
`  Reserved`	24	bslbf
`}`		

test_flag This 8-bit field conveys a set of test flags used to target this application to a specific population of set-top boxes. The usage of this field is governed by the cable operator and not further defined by this specification.

reserved This 24-bit field is reserved by this specification and SHALL be set to 0x000000.

application_priority This field identifies a relative priority between the applications signaled in this service.

- Where there is more than one application with the same Application identification, this priority SHALL be used to determine which application is started.
- Where there are insufficient resources to continue running a set of applications, this priority SHALL be used to determine which applications to terminate.
- The greater the numerical value, the higher the application priority.

initial_resource_locator This field identifies the locator for the initial page resource to be loaded by the receiver to execute the application identified by `application_identifier`. The initial page resource shall be conveyed in an elementary stream that is signaled by the same PMT that signaled the stream containing this EISS table. The format of the locator SHALL comply with the `ebiLocator` structure as defined in [ETV-BIF] Section 11.13 Locator.

For the purposes of this specification, the `lsType` locator type field of the `ebiLocator` structure SHALL be restricted to the following locator type:

- type 4 – URI Locator, wherein the URI SHALL match the URI that is conveyed in the DII message of the initial page resource. This form of locator SHALL be used to reference an initial page resource that conveys a Local Identifier (lid) URI scheme in its associated DII message.

If `application_control_code` equals 0x03 (DESTROY), the user agent SHALL ignore the value conveyed by initial_resource_locator, and in this case the value of the `lsLength` field of `ebiLocator` MAY be zero (0).

private_data This field is defined as private use and is dependent upon the Application Type being signaled. This field SHALL be used to carry an application argument string as identified by [ETV-BIF] Section 9.6.9.7 Application Arguments.

7.2.2 ETV Media Time Descriptor

ETV media time descriptors enable a receiver to maintain a program-specific timeline that can be referenced by a stream event for synchronization of an application to a broadcast program. This descriptor contains a value that allows the receiver to establish a unique time for each point within the program, even when that program is interrupted for advertisements or is joined in progress.

The ETV media time descriptor is defined in Table 11.

Table 11 - ETV Media Time Descriptor Syntax

Syntax	Bits	Mnemonic
`etv_media_time_descriptor() {`		
`  descriptor_tag`	8	uimsbf
`  descriptor_length`	8	uimsbf
`  time_value`	32	uimsbf
`}`		

descriptor_tag This 8-bit integer with value `0xE1` identifies this descriptor.

descriptor_length This 8-bit integer indicates the number of bytes following the `descriptor_length` field.

time_value This field contains the time in milliseconds since the beginning of the current program.

It is not necessary for a program to actually start at time 0, as long as the Stream Event descriptors tied to these time codes accommodate for the actual time values delivered in this stream. Consecutive `time_value`'s SHALL NOT have negative time discontinuities after discounting the interval of time passed between two consecutive `etv_media_time_descriptor'`s. Consecutive `time_value`'s MAY have positive time discontinuities between two consecutive `etv_media_time_descriptor'`s.

The user agent SHALL provide forwards extrapolation of `time_value` for the application between receiving two consecutive `etv_media_time_descriptor'`s.

7.2.3 ETV Stream Event Descriptor

ETV stream event descriptors carry application data to be delivered to an application synchronously with the broadcast event. The ETV stream event descriptor is defined in Table 12.

Table 12 - ETV Stream Event Descriptor Syntax

Syntax	Bits	Mnemonic
`etv_stream_event_descriptor() {`		
`  descriptor_tag,`	8	uimsbf
`  reserved1,`	4	bslbf
`  descriptor_length,`	12	uimsbf
`  time_value`	32	uimsbf
`  for (i=0; i<N; i++) {`		
`  payload_byte`	8	bslbf
`  }`		
`}`		

descriptor_tag This 8-bit integer with value `0xE2` identifies this descriptor.

reserved1 This 4-bit field SHALL be set to `0000`.

descriptor_length This 12-bit integer indicates the number of bytes following the `descriptor_length` field.

time_value This field indicates the time at which the event SHOULD be delivered to the application on the receiver. There will be some inevitable delay based on the processing power of the receiver. If `time_value` is equal to 0, the event SHALL be delivered immediately. The semantics related to this field are described in Section 7.2.2.

payload_byte This field contains application-dependent data. The semantics of this field are application type-dependent. This field SHALL be interpreted according to the format described by [ETV-BIF] section 11.17.1 Serialized Trigger.

7.3 Synchronization in Analog Services

Synchronization and signaling for analog services is outside the scope of this document.

8 CARRIAGE OF ETV APPLICATION RESOURCE DATA

This section describes the carriage of ETV application resource data within an MPEG-2 Transport Stream. A common portable format must be established that may be interpreted by all receivers, and which can be generated by all authoring and packaging tools.

8.1 DSM-CC Data Carousel

If the Elementary Stream that carries ETV Resources is signaled with a `stream_type` of `0x0B` (IEC 13818-6 Type B - DSM-CC Data Carousel [IEC 13818-6]), the application resource data is carried in a DSM-CC Data Carousel as defined in [IEC 13818-6]. No constraints are placed on the Data Carousel by this specification.

The contents of sections 7 and 9 of [IEC 13818-6] are hereby incorporated into this specification; and, for the purposes of the OpenCable Contribution Agreement, SHALL be considered a “Contribution” to this specification, subject to the IPR terms and conditions (including each signatory's opportunity to provide notice) of the OpenCable Contribution Agreement.

The following fields of the DownloadInfoIndication (DII) message are further specified by this specification:

moduleInfoByte The moduleInfoByte fields of the DII message MAY specify the *abs_path* component of the Local Identifier (lid:) URI Scheme [SMPTE343M] to be used to locate this module when used by a URI Locator as defined by [ETV-BIF] section 11.13.5. If moduleInfoByte fields are specified, then the user agent SHALL provide a mapping to this module for lid: URI Locators that reference this module using *authority* and *abs_path*. The moduleInfoByte fields contain the *abs_path* structure as defined in Table 13.

Table 13 - abs_path

Syntax	Bits	Mnemonic
`abs_path() {`		
`  abs_path_length`	8	uimsbf
`  for (i=0; i<N; i++) {`		
`  abs_path_byte`	8	bslbf
`  }`		
`}`		

abs_path_length This 8-bit integer indicates the number of bytes of the abs_path structure following the abs_path_length field.

abs_path_byte This field contains the bytes of the *abs_path* component of the lid: URI scheme.

privateDataByte The privateDataByte fields of the DII message MAY specify the *authority* component of the Local Identifier (lid:) URI Scheme [SMPTE343M] to be used to locate the resources identified by this DII message when referenced by an lid: URI Locator as defined by [ETV-BIF] section 11.13.5. The privateDataByte fields contain the *authority* structure as defined in Table 14.

Table 14 - authority

Syntax	Bits	Mnemonic
`authority() {`		
`  authority_length`	8	uimsbf
`  for (i=0; i<N; i++) {`		
`  authority_byte`	8	bslbf
`  }`		
`}`		

authority_length This 8-bit integer indicates the number of bytes of the *authority* structure following the `authority_length` field.

authority_byte This field contains the bytes of the *authority* component of the lid: URI scheme. Resources for a single authority SHALL NOT be conveyed on more than one data carousel in a given elementary stream. Resources for different authorities SHALL be conveyed on different data carousels in a given elementary stream. In other words, a DII SHALL map all resources that are being conveyed for a given authority in the elementary stream at any given moment.

8.2 Alternate Constrained Data Carousels

If the Elementary Stream that carries ETV Resources is signaled with a `stream_type` of `0xC0` (DCII Text Message), ETV Resources SHALL be encapsulated within a constrained data carousel as defined in Table 15. This constrained data carousel carries a DSM-CC_Section as defined in [IEC 13818-6]. The primary constraint imposed by this format is the limitation that each `dc2_data_carousel_section` is limited in size to less than 1 KB.

Table 15 - DCII Data Carousel Message Syntax

Syntax	Bits	Mnemonic
`dc2_data_carousel_section() {`		
`  table_id`	8	uimsbf
`  section_syntax_indicator`	1	bslbf
`  reserved1`	3	bslbf
`  section_length`	12	uimsbf
`  filter_info`	16	uimsbf
`  reserved2`	8	bslbf
`  for (i=0; i<N; i++) {`		
`  dsmcc_section()`		
`  }`		
`  CRC32`	32	rpchof
`}`		

table_id This 8-bit integer with value `0xE3` or `0xE4` identifies this descriptor. If the enclosed dsmcc_section carries a DII message, this field SHALL convey the value 0xE3. If the enclosed dsmcc_section carries a DownloadDataBlock (DDB) message, this field SHALL convey the value 0xE4.

section_syntax_indicator The `section_syntax_indicator` is a 1-bit field which SHALL be set to 0.

reserved1 This 3-bit field is reserved by this specification and SHALL be set to ‘100’.

section_length	This 12-bit field specifies the number of bytes in the section starting immediately following the `section_length` field. The value in this field SHALL not exceed 1021, indicating that the encapsulated `dsmcc_section` SHALL have a maximum length of 1014 bytes.
filter_info	This 16-bit field is intended to accommodate hardware filtering of messages. If the enclosed dsmcc_section carries a DII message, this field SHALL convey the value 0xFBFB. If the enclosed dsmcc_section carries a DDB message, this field SHALL convey a copy of the moduleId field of the conveyed DSMCC DDB message. The set of values for the moduleId SHALL be limited to within the range 0x0001 and 0xFBEF. This enables a receiver to set hardware filters on all DSMCC DII control messages and specific download data modules in dsmcc_section sections.
reserved2	This 8 bit field SHALL be set to `0x00`.
dsmcc_section	This field carries a DSM-CC_Section as defined in [IEC 13818-6], table 9-2. When a DSM-CC section is encapsulated by a `dc2_data_carousel_section` the maximum length of that DSM-CC section is 1014 bytes.
CRC-32	This field SHALL be set as defined in [IEC 13818-1], Annex B.

9 APPLICATION SIGNALING AND SYNCHRONIZATION FOR LIMITED CAPABILITY DEVICES

9.1 Introduction

ETV applications will be deployed on a wide range of receivers, including devices such as the Motorola DCT-2000. Some devices do not have the resources necessary to support advanced signaling techniques. Programmers and network operators may choose to simultaneously broadcast both messaging types in order to target the widest range of devices; however, all devices must be capable of reading the base signaling stream.

The critical resource constraint of limited-capability devices is the number of PID filters, which has required the combination of several elements into the Integrated Signaling Stream as described in Section 7 of this specification.

An overall review of the descriptor elements introduced in this specification is illustrated in Figure 3:

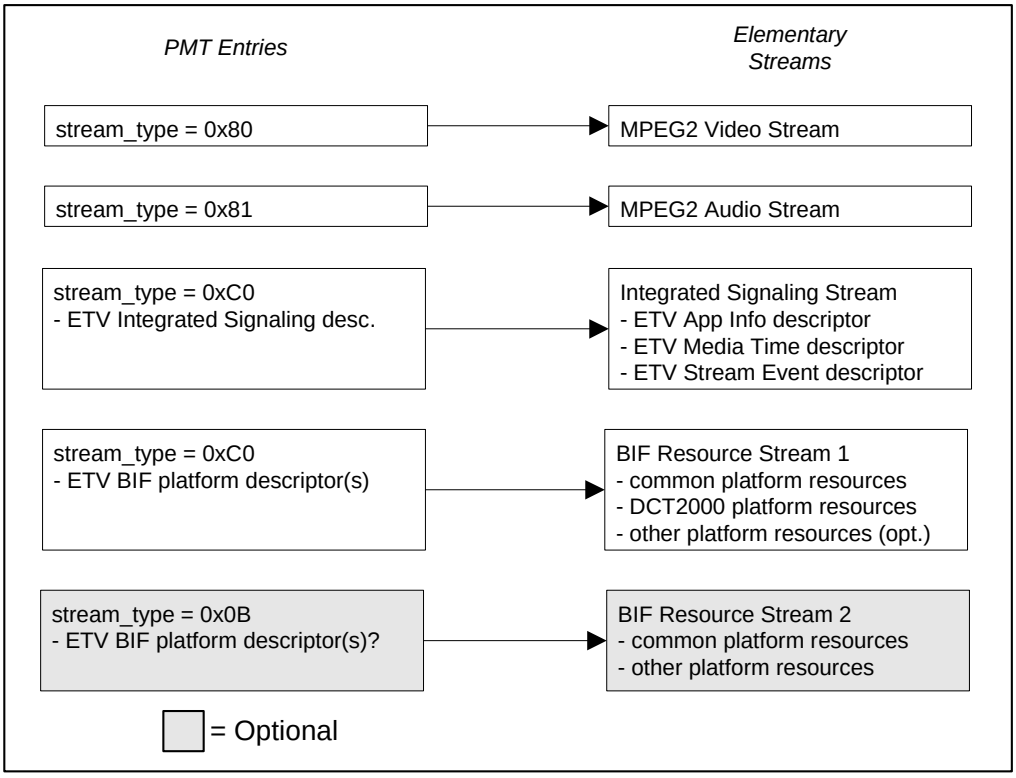

Figure 3 - PMT Signaling Motorola DCT-2000 Specific Behaviors

The User Agent running on a DCT-2000 SHALL read the first four PMT entries in the above table. The ETV Integrated Signaling Descriptor in the third PMT entry indicates that the associated elementary stream carries a signaling stream that includes an AIT-like Application Information descriptor, the Media Time descriptors, and any Stream Event descriptors.

The ETV-BIF Platform descriptor in the fourth entry indicates that this elementary stream carries ETV-BIF resources for the specified platforms.

If one or more elementary streams contains ETV-BIF resources targeted to a DCT-2000 receiver, those streams will have a `stream_type` of `0xC0` (DCII Text Message). Those resources that are not required to be processed by a DCT-2000 MAY be signaled with a `stream_type` of `0xC0` or `0x0B` (DSM-CC Data Carousel).

9.2 All other Set-Top Specific Behaviors

User Agents running on all other receivers SHALL read the first three PMT entries, just as the DCT-2000. The Application Information descriptor in the EISS will contain an `application_identifier`, which, in this case, MAY reflect the presence of resources in either the fourth or fifth stream (or both) above. The User Agent SHALL read each PMT entry to find the resources most suitable for the given hardware and/or software platform and load from those streams as required.

9.3 OpenCable Host Specific Behaviors

No specific signaling or behaviors have been identified for OpenCable hosts.

Appendix I Revision History

The following ECNs were incorporated into OC-SP-ETV-AM-I02-050726:

Number	Description	Date
ETV-AM-N-05.0785-2	Support for single ETV data PID	7/13/05
ETV-AM-N-05.0786-2	Correction to filter_info values in App Resource stream	7/13/05
ETV-AM-N-05.0804-1	Improved specification for time_value field of etv_media_time_descriptor	7/20/05

The following ECNs were incorporated into OC-SP-ETV-AM1.0-I03-060714:

Number	Description	Date
ETV-AM-N-05.0828-3	Collected changes for ETV AM	1/16/06
ETV-AM1.0-N-06.0861-3	Improved support for enhanced Ads	4/3/06

The following ECN was incorporated into OC-SP-ETV-AM1.0-I04-070921:

Number	Description	Date
ETV-AM1.0-N-06.0963-2	Collected and editorial changes	4/10/07

The following ECNs were incorporated into OC-SP-ETV-AM1.0-I05-091125:

Number	Description	Date
ETV-AM1.0-N-08.1196-1	Clarify Hardware/Software Version field descriptions	4/1/08
ETV-AM1.0-N-09.1401-2	Application Priority Clarification	11/25/09
ETV-AM1.0-N-09.1429-2	Initial Page Resource location clarification	11/25/09