

---

# SMPTE STABLE DOCUMENT

---



**The attached SMPTE Engineering Document has been declared “Stable” by the controlling Technology Committee.**

**The SMPTE Operations Manual for Standards states:**

*A document should be stabilized if it is believed to be substantially correct, does not contain harmful or misleading recommendations, may still be relevant to equipment or practices in use, is stable, but does not represent current technology, and need not be subject to future reviews.*

*A Stable document shall still be made available and offered for sale by the Society, but it shall be prefaced by a cover page explaining its current status.*

*At any time, a Technology Committee may revise, amend, or otherwise initiate a new Project on a Stable document.*

**A Stable document is “In Force”, and not deprecated or withdrawn.**

**\* \* \* \* \***

**Note:**

**SMPTE “Stable” documents were previously described as “Archived” and the attached document may be marked as “Archived”. The status of a SMPTE document described as “Archived” is exactly as described above for a “Stable” document.**

**Stable documents may not adhere to the latest style and format of SMPTE documents, or to current usage of normative language. Suitable care should be taken in interpretation.**

# SMPTE STANDARD

## for Television — Declarative Data Essence — Content Level 1



Page 1 of 20 pages

### Table of contents

- 1 Scope
- 2 Normative references
- 3 Introduction
- 4 Content formats
- 5 Transport types
- Annex A JFIF guidelines for JPEG
- Annex B Examples of integrating television with web pages
- Annex C Using enhanced television
- Annex D Glossary
- Annex E Receiver trigger behavior
- Annex F Security model
- Annex G Bibliography

## 1 Scope

This standard defines a standard for the authoring of declarative data content intended to be combined primarily with video and/or audio services, and distributed to data-capable television signal receivers. Declarative content is generally nonprocedural, and most commonly in the form of HTML. However, procedural scripting is also defined.

## 2 Normative references

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

EIA-746-A-1988, Transport of Internet Uniform Resource Locator (URL) Information Using TEXT-2 (T-2) Service

SMPTE 343M-2002, Television — Declarative Data Essence — Local Identifier (lid:) URI Scheme

SMPTE 366M-2002, Television — Declarative Data Essence — Document Object Model Level 0 (DOM-0) and Related Object Environment

IETF RFC 1952, GZIP File Format Specification Version 4.3

IETF RFC 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types

IETF RFC 2068, Hypertext Transfer Protocol — HTTP/1.1

IETF RFC 2110, MIME E-mail Encapsulation of Aggregate Documents, Such as HTML (MHTML)

IETF RFC 2318, The Text/CSS Media Type

IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax

IETF RFC 2838, Uniform Resource Identifiers for Television Broadcasts

ISO 8859-1:1998, 8-Bit Single-Byte Coded Graphic Character Sets — Part 1: Latin Alphabet No. 1

ISO/IEC 10918-1:1994, Information Technology — Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines

ISO/IEC 11578:1996, Information Technology — Open Systems Interconnection — Remote Procedure Call (RPC), Annex A, Universal Unique Identifier

ISO/IEC 16262:1998, Information Technology —ECMAScript Language Specification

W3C Recommendation, Cascading Style Sheets, Level 1 (CSS1)

W3C Recommendation, HyperText Markup Language, version 4.0

W3C Recommendation, PNG (Portable Network Graphics) Specification

IANA Media Types, application/tve-trigger, (<http://www.isi.edu/in-notes/iana/assignments/media-types/media-types>), Registration of MIME media type application/tve-trigger (<http://www.isi.edu/in-notes/iana/assignments/media-types/application/tve-trigger>)

Object Management Group (OMG) CORBA/IIOP 2.3.1, The Common Object Request Broker: Architecture and Specification, Section 3, OMG IDL Syntax and Semantics

### 3 Introduction

This document defines a standard for the authoring of declarative data content intended to be combined with video and audio services and distributed eventually to data-capable television receivers.

It is assumed that the reader is familiar with all the normative references; basic concepts are not discussed in this standard.

This specification for enhanced television programming uses existing Internet technologies. It delivers enhanced television programming over both analog and digital video systems using terrestrial, cable, satellite, and Internet networks. The specification can be used in both one-way broadcast and two-way video systems, and is designed to be compatible with all international standards for both analog and digital video systems.

A central design point is to use existing standards wherever possible and to minimize the creation of new specifications. Existing web standards, with only minimal extensions for television integration, provide a rich set of capabilities for building enhanced television content in today's marketplace. This standard references full existing specifications for HTML, ECMAScript, DOM, CSS, and media types as the basis of the content specification. The specification is not a limit on what content can be sent, but rather provides a common set of capabilities so that content developers can author content once and play on the maximum number of receivers.

Another key design goal is to provide a single solution that would work on a wide variety of networks. This design is capable of running on both analog and digital video systems as well as networks with no video at all. The specification also supports transmission across terrestrial (over the air), cable, and satellite systems as well as over the Internet. In addition, it will also bridge between networks — for example, data on an analog terrestrial broadcast must easily bridge to a digital cable system. This design goal was achieved through the definition of a transport-independent content format. This document defines two transports — one for broadcast data and one for data pulled through a return path.

Although this specification has the capability to run on any video network, a complete specification requires a specific binding to each video network standard in order to ensure true interoperability. There are many roles in the production and delivery of television enhancements. This standard refers to three key roles: content creator, transport operator, and receiver. The content creator originates the content components of the enhancement including graphics, layout, interaction, and triggers. The transport operator runs a video delivery infrastructure (terrestrial, cable, satellite, or other) that includes a transport for the data. The receiver is a hardware and software implementation (television, set-top box, or personal computer) that decodes and plays this content. A particular group or company may participate as one, two, or all three of these roles.

## 4 Content formats

The foundation for the declarative content is existing web standards. Mandatory support is required for the following standard specifications:

- HTML 4.0 ([W3C HyperText Markup Language]; all three document type definitions);
- CSS1 (W3C Cascading Style Sheets, Level 1);
- ECMAScript (ISO/IEC 16262);
- DOM-0 (SMPTE 366M).

### 4.1 Fonts

The following two fonts are "default" and should be expected by authors to reside in the receiver for Latin character (ISO/IEC 8859-1) markets:

- Times, or another sans-serif font with equivalent metrics;
- Monospace.

The different kinds of these basic font families are defined in table 1.

The CSS keyword font size is defined in table 2.

**Table 1 – Basic font families**

| Family    | Size (pixels) | Weight | Style  |
|-----------|---------------|--------|--------|
| Tieresias | 12            | Normal | Normal |
| Tieresias | 24            | Normal | Normal |
| Tieresias | 18            | Normal | Normal |
| Tieresias | 14            | Normal | Normal |
| Tieresias | 10            | Normal | Normal |
| Tieresias | 9             | Normal | Normal |
| Tieresias | 12            | Bold   | Normal |
| Tieresias | 24            | Bold   | Normal |
| Tieresias | 18            | Bold   | Normal |
| Tieresias | 14            | Bold   | Normal |
| Tieresias | 10            | Bold   | Normal |
| Tieresias | 9             | Bold   | Normal |
| Tieresias | 12            | Normal | Italic |
| Monospace | 12            | Normal | Normal |

**Table 2 – CSS font size**

| Font size name | Height (mm) |
|----------------|-------------|
| xx-small       | 4           |
| x-small        | 6           |
| small          | 8           |
| medium         | 12          |
| large          | 18          |
| x-large        | 25          |
| xx-large       | 40          |

## 4.2 ECMAScript support

Authors shall assume ECMAScript support by receivers compliant with SMPTE 366M. ECMAScript syntax should be both parsed and executed. All native objects and methods should be fully implemented and all language syntax and semantics supported. If authors use script objects, methods, or attributes not defined in DOM-0, such as to target a superset of DOM-0 supported by a specific browser implementation, it is the author's responsibility to insert conditional logic that will only present that script for parsing to the intended browser. Authors are advised of possible implementation limitations as follows:

- Array sizes may be limited to 32,768 elements;
- The number data type may be implemented as a 32-bit floating point value;
- The method, Object.prototype, may not be supported.

### 4.2.1 Cookie support

Authors may count on 1kB for session cookies. Cookie support is not assumed to be persistent when a receiver is turned off.

## 4.3 Content type support

Because the architecture supports one-way broadcast of data, content creators cannot customize the content for each receiver as they do today with two-way HTTP. The following is a base profile of supported MIME types that may be included in DDE content:

- text/html (HTML 4.0);
- text/plain;
- text/css (CSS1 only);
- image/png (no progressive encoding);

- image/jpeg (no progressive encoding);
- audio/basic.

#### 4.3.1 Image/png

All image/png content is expected to be cached. All possible syntax shall be permitted in the content. All features are expected to be decoded and rendered except as follows:

- Non-square pixels;
- Gamma correction;
- Chroma correction;
- Progressive display

#### 4.3.2 Image/jpeg

All encoding formats are permitted in the content. However, it is only expected that full JFIF (annex A) guidelines may be decoded and rendered. The thumbnail image may be ignored.

### 4.4 Triggers

Triggers are real-time events delivered for the enhanced television program.

Note that receiver implementations may set their own policy for allowing users to turn on or off enhanced television content, and can use trigger arrival as a signal to notify users of enhanced content availability. They are also free to decide how to turn on enhancements and how to enable the user to choose among enhancements.

Triggers always include a URL, and may optionally also include a human-readable name, an expiration date, and a script. Triggers that include a *name* attribute may be used to initiate an enhancement either automatically, or with user confirmation. If an enhancement is not currently loaded, the expiration has not been reached, and the trigger contains a *script* attribute, then the script is executed through the trigger receiver object when the page is loaded, and after all OnLoad events have fired for the enhancement. If a replaceable enhancement is currently loaded and a trigger is received whose URL does not match the currently loaded top-level page, the trigger expiration has not been reached, the trigger includes a *name* attribute and a *script* attribute; then the new enhancement can be offered to the user or automatically loaded. If the top-level page of the new enhancement is loaded, the script attribute will be evaluated and executed after loading is completed. If an enhancement is currently loaded and a trigger with a script attribute is received with a URL that matches the current top-level page, and the expiration has not been reached, the script attribute will be immediately evaluated and executed. If duplicate triggers with script attributes are received in sequence, script attributes will be evaluated and executed for each instance. The initial top-level page for that enhancement is indicated by the URL in that trigger. Triggers that do not include a *name* attribute are not intended to initiate an enhancement, but should be processed only as events which affect (through the *script* attribute) enhancements that are currently active. If the URL matches the current top-level page and the expiration has not been reached, the script is executed through the trigger receiver object. When testing for a match, parameters and fragment identifiers (i.e., characters in the URL including and following the first "?" or "#" character) in a URL are ignored.

Triggers are text based, and their syntax follows the basic format of the EIA-746-A standard (7-bit ASCII, the high-order bit of the first byte must be "0").

NOTE – The triggers follow the syntax of EIA-746-A, but may be transported in ways appropriate for the transport, such as multicast IP packets for example, rather than using the EIA/CEA-608-B system.

All triggers are text-based and must begin with ASCII '<'. All other values for the first byte are reserved. These reserved values may be used in the future to signal additional nontext based messages. Receivers may ignore any trigger that does not begin with the '<' in the first byte.

The general format for triggers (consistent with EIA-746-A) is a required URL followed by zero or more attribute/value pairs and an optional checksum:

```
<url> [attr1:val1][attr2:val2]...[attrn:valn][checksum]
```

The requirement to provide a checksum is transport binding specific. If required by the transport binding, the checksum must come at the end of the trigger. It is required for transport A and not required for transport B.

- Character set: All characters are based on ISO/IEC 8859-1 character set (also known as Latin-1 and compatible with US-ASCII) in the range 0x20 and 0x7e. Any need for characters outside of this range (or excluded by attribute limits below) must be encoded using the standard Internet URL mechanism of the percent character ("%") followed by the two-digit hexadecimal value of the character in ISO/IEC 8859-1.

- Trigger: The trigger begins with a required URL:

<url>: The URL is enclosed in angle brackets (e.g., <http://xyz.com/fun.html>). Although any URL can be sent in this syntax, content level 1 only requires support for http: and lid: URL schemes.

The following attribute/value pairs are defined:

**[name:string]:** The name attribute provides a readable text description (e.g., [name:Find Out More]). The string is any string of characters between 0x20 and 0x7e except square brackets (0x5b and 0x5d) and angle brackets (0x3c and 0x3e). The name attribute can be abbreviated as the single letter "n" (e.g., [n:Find Out More]).

**[expires:time]:** The expires attribute provides an expiration date, after which the link is no longer valid (e.g., [expires:19971223]). The time conforms to the ISO 8601 standard, except that it is assumed to be UTC unless the time zone is specified. A recommended usage is the form *yyyymmddThhmmss*, where the capital letter "T" separates the date from the time. It is possible to shorten the time string by reducing the resolution. For example, *yyyymmddThhmm* (no seconds specified) is valid, as is simply *yyyymmdd* (no time specified at all). When no time is specified, expiration is at the beginning of the specified day. The expires attribute can be abbreviated as the single letter "e" (e.g., [e:19971223]).

**[script:string]:** The script attribute provides a script to execute within the context of the page containing the trigger receiver object (e.g., [script:shownews( )]). The string is an ECMAScript statement. The form of statement shall be in accordance with the StatementList nonterminal of ECMAScript. The script attribute can be abbreviated as the single letter "s" (e.g., [s:shownews( )]). An example of a script attribute used to navigate a frame within a page to a new URL:

```
[script:frame1.location.href="http://atv.com/f1"]
```

**[tve:version]:** The tve attribute indicates to the receiver that the content described in the trigger is conformant to the content specification level, *version*. For example, [tve:1.0]. The tve: attribute can be abbreviated as the single letter "v". The *version* can be abbreviated to a single digit when the *version* ends in ".0" (e.g., [v:1] is the same as [tve:1.0]).

The optional checksum must come at the end of the trigger.

NOTE – EIA-746-A requires the inclusion of a checksum to ensure data integrity over line 21 bindings. In other bindings, such as IP, this may not be necessary, and is not required.

**[checksum]:** The checksum is provided to detect data corruption. To compute the checksum, adjacent characters in the string (starting with the left angle bracket) are paired to form 16-bit integers; if there is an odd number of characters, the final character is paired with a byte of zeros. The checksum is computed so that the ones complement of all of these 16-bit integers plus the checksum equals the 16-bit integer with all 1 bits (0 in ones