
**Information technology — High
efficiency coding and media delivery
in heterogeneous environments —**

**Part 11:
MPEG Media Transport Composition
Information**

*Technologies de l'information — Codage à haute efficacité et livraison
des médias dans des environnements hétérogènes —*

*Partie 11: Informations de composition pour le transport des médias
MPEG*

IECNORM.COM : Click to view the full PDF of ISO/IEC 23008-11:2015



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

	Page
Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Symbols and abbreviated terms	1
4 Overview	1
5 Composition Information	2
5.1 Introduction	2
5.2 Structure of the CI document	3
5.2.1 Overview	3
5.3 The attributes of a CI document	4
5.3.1 The <i>begin</i> attribute	4
5.3.2 The <i>end</i> attribute	4
5.3.3 The <i>dur</i> attribute	4
5.3.4 The <i>clipBegin</i> attribute	5
5.3.5 The <i>clipEnd</i> attribute	5
5.3.6 The <i>refDiv</i> attribute	5
5.3.7 The <i>style</i> attribute	5
5.3.8 The <i>xlink:href</i> attributes	5
5.3.9 The <i>xlink:actuate</i> attributes	5
5.3.10 The <i>viewRole</i> attributes	5
5.3.11 The <i>isDependent</i> attributes	5
5.3.12 The <i>depld</i> attributes	6
5.3.13 The <i>refld</i> attribute	6
5.3.14 The <i>mediaSrc</i> attribute	6
5.3.15 The <i>obsolete</i> attribute	7
5.3.16 The <i>obsoleteTime</i> attribute	7
5.4 The elements of a CI document	8
5.4.1 The CI element	8
5.4.2 The reference element	8
5.4.3 The view element	9
5.4.4 The area element	10
5.4.5 The MediaSync element	11
5.4.6 The sourceList element	12
6 HTML data attributes	12
6.1 Introduction	12
6.2 Attributes	12
6.2.1 data-version attribute	12
6.2.2 data-ci attribute	12
7 Update Mechanism	13
7.1 Introduction	13
Annex A (informative) XML Schema of Composition Information	14
Annex B (informative) Examples of CI document	17
Bibliography	34

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: [Foreword — Supplementary information](#).

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 23008 consists of the following parts under the general title *Information technology — High efficiency coding and media delivery in heterogeneous environments*:

- *Part 1: MPEG media transport (MMT)*
- *Part 2: High efficiency video coding (HEVC)*
- *Part 3: 3D Audio*
- *Part 5: HEVC Conformance testing and reference software*
- *Part 8: Conformance Specification for HEVC*
- *Part 10: MPEG Media Transport Forward Error Correction (FEC) codes*
- *Part 11: MPEG Media Transport Composition Information*
- *Part 12: Image file format*
- *Part 13: MMT Implementation guidelines*

Introduction

This part of ISO/IEC 23008 specifies technologies for the delivery of coded media data for multimedia services over concatenation of heterogeneous packet-based network segments, including bidirectional IP networks and unidirectional digital broadcasting networks. In this part of ISO/IEC 23008, “coded media data” includes both timed audiovisual media data requiring synchronized decoding and presentation of each specific unit of data at a designated time and non-timed data that could be decoded and presented at an arbitrary time based on the context of the service or the user interaction.

MMT is designed under the assumption that the coded media data will be delivered through a packet-based delivery network. Several characteristics of such delivery environments have been taken into consideration, such as non-constant end-to-end delivery delay of each packet from the sending entity to the receiving entity and means to distinguish signaling messages from the media data provided by the underlying network.

For efficient and effective delivery of coded media data over heterogeneous packet-based delivery networks, this part of ISO/IEC 23008 provides the following elements:

- logical model to construct content composed of components from various sources (e.g. content for mash-up applications);
- structure of data conveying information about the coded media data for processing by the delivery layer (e.g. packetization and adaptation);
- packetization method and the structure of the packet to deliver media content over packet-based delivery networks supporting hybrid multichannel delivery that is agnostic to the specific type of media or coding method;
- format of signaling messages to manage the presentation and delivery of media content;
- format of information to be exchanged across layers of the delivery network to facilitate cross layer communication.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23008-11:2015

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 11: MPEG Media Transport Composition Information

1 Scope

This part of ISO/IEC 23008 specifies MPEG Composition Information (CI), a method describing composition information of media for delivery of multimedia services over packet-based heterogeneous networks.

The technologies for composition function specify the method associating content delivered in the format defined in this part of ISO/IEC 23008 to the presentation and the method representing synchronization between timed and non-timed content.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

W3C HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Candidate Recommendation 17 December 2012

3 Symbols and abbreviated terms

AU	access unit
CI	composition information
MMT	MPEG media transport
MPU	media processing unit
SVC	scalable video coding
URL	uniform resource locator
URN	uniform resource name
XML	extensible mark-up language

4 Overview

This part of ISO/IEC 23008 defines a Composition Information (CI) layer that controls the temporal and spatial layout of media. The composition layer is currently specified using HTML5 and XML technologies.

The composition layer defines the presentation aspects of a multimedia presentation such as an MMT Package. The composition layer in this part of ISO/IEC 23008 addresses delivery of synchronization information of content that is composed of various content components and where multiple device screens may be used to present the content. Some parts of the content component may be replaced by others based on the context of the user, such as location or personal preferences. Presentation time of

some components may not be assigned when the content is encapsulated but needs only to be decided when the content is actually consumed.

A multimedia service may use any subset of the tools defined in this part of ISO/IEC 23008 according to their specific needs. Interfaces to other protocols and standards are either defined by this part of ISO/IEC 23008 or may be defined elsewhere.

5 Composition Information

5.1 Introduction

The Composition layer provides information on temporal relationships among the media content of a presentation to complement the associated HTML5 document, since the associated HTML5 document only provides initial information on spatial relationships among media elements. The composition layer also provides information on relationships between the media data (e.g. the MMT Assets) and possibly on how to optimally consume them on multiple screens. The Composition Information (CI) is represented as an XML document (CI document) providing descriptive representation of such temporal relationship information, thus enabling various ways to implement the composition layer (including implementations that do not use any scripts processing engine.)

To support descriptive representation of temporal relationships among media data (such as MMT assets or MPUs) and mapping of media data to be consumed on multiple devices, the composition layer defines several elements and attributes as follows:

- Association of HTML5 media elements and media data as resources.
- Temporal information to presentation time of the media data.
- Designation of certain parts of the presentation to be presented on a particular screen in a multi-screen environment.

A CI processing engine is responsible for fetching the CI file and the HTML5 file (and any other referenced files) and processing the CI information to control the presentation accordingly.

The HTML5 file is typically parsed into a Document Object Model (DOM) tree and stored in memory. The CI processing engine applies changes to the DOM at specified time according to the information contained in the CI file. The DOM nodes/elements are referenced using their identifiers or possibly using a certain pattern (e.g. provided through jQuery selectors).

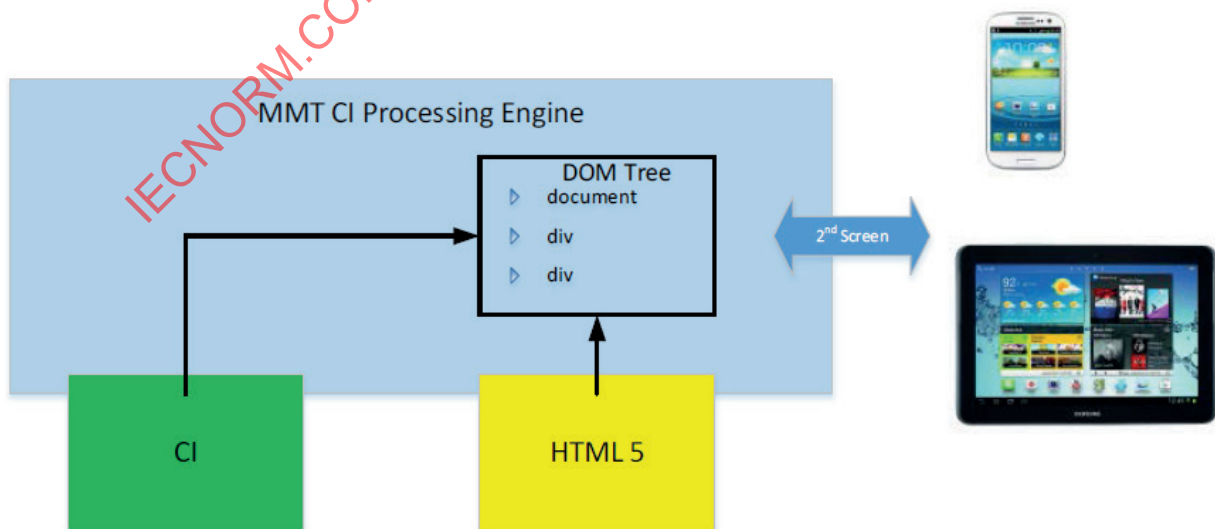


Figure 1 — CI Layer

5.2 Structure of the CI document

5.2.1 Overview

The CI document is defined as an XML document with the XML schema provided in [Annex A](#). The CI document is defined with several elements referring the elements defined in the HTML5 document by their IDs as shown in [Figure 2](#).

The HTML5 document shall include HTML5 media elements, which maybe any of **video**, **audio**, **img**, or **track elements** together with their associated spatial information. A reference to the CI document may be included in the HTML5 document, typically as part of the Javascript that launches the CI parser. Spatial information of the media elements shall be used to present the corresponding media data (such as the MMT Assets). The **body** element of the HTML5 document shall include more than one spatial area (i.e. **div** element), if partial consumption of the presentation on multiple screens is to be allowed.

To efficiently represent temporal relationships among media data to be synchronously consumed on multiple screens that are independently processing a presentation, the CI document defines in a hierarchical structure the concepts of Area and View. An Area in the CI document represents a spatial region defined by a **div** element in the HTML5 document, which may include one or more HTML5 media elements. A View represents a set of Areas to be consumed on a single screen.

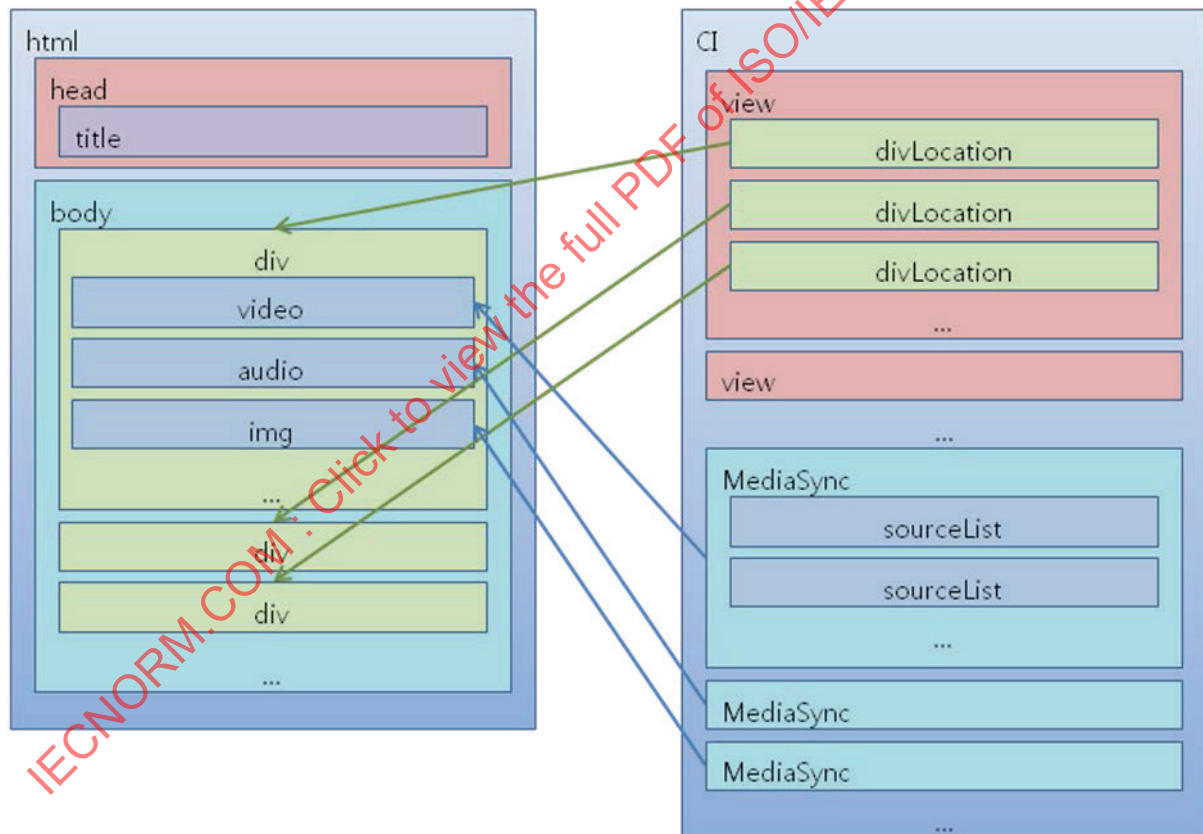


Figure 2 — Structure of CI document and its relationship with HTML5 document

The CI document provides temporal information about Views, Areas and media data. The root element of CI document is the **CI** element, which may contain elements such as **view** and **MediaSync** element. The roles of these elements are as follows:

- **view**: provides temporal information about the spatial changes of a View and its Areas.
- **MediaSync**: provides temporal information about the presentation of media data (e.g. MMT assets).

The initial Areas in a View are generated by the **div** element in the HTML5 document, and the spatial changes of the initial Areas in the View are provided by the **area** elements of the CI document. These two elements allow the spatial location of Areas to be dynamically changed and updated without reloading the HTML5 document.

In MMT delivery, the CI data may be partitioned into multiple subsets that reference each other using the XLink framework and where each subset of the CI may be delivered separately for example by using the MPI signaling message (see ISO/IEC 23008-1 sub-clause 5.3.8 and 5.3.9).

Figure 3 shows an example representing the relationship among media data (MMT assets), Areas and View.

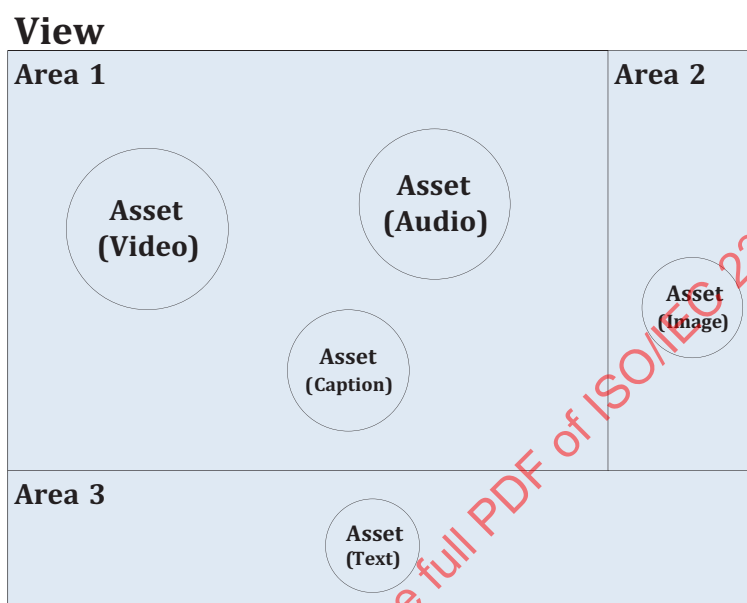


Figure 3 — Example of Asset/Area/View

CI uses the timing and synchronization module of SMIL [W3C SMIL 3.0]. For CI consumption the timing and synchronization module shall be supported.

5.3 The attributes of a CI document

5.3.1 The *begin* attribute

The *begin* attribute indicates when the corresponding command has to be performed by the user agent, e.g. when particular media data needs to be rendered by an HTML5 media element or when a change to the spatial layout of the presentation has to be performed. The *begin* attribute is defined in section 5.4.3 of [W3C SMIL3.0]. All defined synchbases defined by [W3C SMIL 3.0] shall be supported.

5.3.2 The *end* attribute

The *end* attribute indicates when the corresponding CI action shall end. For media data it indicates when the media data playback stops. When the parent element is a view element, the agent shall completely remove the referenced **div** element from the DOM at the indicated end time. When not present for media data, the end time shall be the time when the playback of the media data ends. The *end* attribute is defined in section 5.4.3 of [W3C SMIL3.0].

5.3.3 The *dur* attribute

The *dur* attribute indicates the duration of the corresponding action, which is equivalent to the end time – the begin time. The *dur* attribute shall not be present when the corresponding action is about spatial layout. The *dur* attribute is defined in section 5.4.3 of [W3C SMIL3.0].

5.3.4 The *clipBegin* attribute

The *clipBegin* attribute indicates the beginning of a time interval of a continuous media element by offset from the start of the media resource. The *clipBegin* attribute is defined in section 4.8.1 of [W3C SMIL3.0]. For example, a *clipBegin*="10s" attribute would indicate that the first 10 seconds of the media resource will not be played back.

5.3.5 The *clipEnd* attribute

The *clipEnd* attribute indicates the end of a time interval of a continuous media element by offset from the start of the media. The *clipEnd* attribute is defined in section 4.8.1 of [W3C SMIL3.0].

5.3.6 The *refDiv* attribute

The *refDiv* attribute describes an ID of **div** element, which is used for linking to its **area** element. If the referenced **div** element is not present in the HTML5, then a new div element is created and added to the DOM at the indicated **begin** time.

5.3.7 The *style* attribute

The *style* attribute shall be used to indicate any spatial layout or style modifications to the referenced **div** element. The style attribute shall conform to the CSS Style Attributes.

5.3.8 The *xlink:href* attributes

The *xlink:href* attribute provides the data that allows an XLink application to find a subset of CI data. The value of the href attribute shall be an URI reference as defined in [RFC 2396] and shall be provided.

5.3.9 The *xlink:actuate* attributes

The *xlink:actuate* attribute defines the desired timing of traversal linked resources. If a value is supplied for an actuate attribute, it must be one of following values.

- *onLoad*: linking a subset of CI data immediately on loading of the current CI document.
- *onRequest*: linking a subset of CI data when they are available or on user request (on-demand).

5.3.10 The *viewRole* attributes

The *viewRole* attribute indicates the role of the **view** element. Several **view** elements can be contained in one CI document, and this attribute can be used to distinguish each of them. The following table defines the meaning of the possible values for the *viewRole* attribute.

Table 1 — the *viewRole* for multi-screen presentation

Value	Description
Default	Indicates that this view is meant to be consumed on the main screen without required interaction with other screens.
Multiple	Indicates that the view is meant to be consumed in a multi-screen environment.
Receptible	Indicates that the view is a placeholder to be used to receive content in a multi-screen presentation.

5.3.11 The *isDependent* attributes

The *isDependent* attribute indicates whether the media element depends on another media element or not, in order to be properly decoded and presented. If the *isDependent* attribute is "true", the media element depends on other media elements that are listed in *depld*.

5.3.12 The *depId* attributes

The *depId* attribute is a whitespace-separated list of *refId* of media elements on which the current media element depends on. The *depId* attribute exists if and only if the *isDependent* attribute is set to “true”.

5.3.13 The *refId* attribute

The *refId* attribute identifies a media element (**video**, **audio**, **img**, **track**, **object** element of [W3C HTML5]). This attribute is used for linking with a **MediaSync** element.

5.3.14 The *mediaSrc* attribute

The *mediaSrc* attribute indicates the address of media data (e.g. a particular MPU of an MMT Asset). The *mediaSrc* attribute MAY also refer to an MPD. When an MPD is referenced the MPD@availabilityStartTime shall be ignored and the time defined in the *begin* attribute is used instead to determine when to start playback of the Presentation. When the *mediaSrc* attribute refers an MMT MPU, the URL format is provided in ISO/IEC 23009-1.

```
<!-- mmtci:begin Type -->
<xsd:simpleType name="beginType">
  <xsd:list itemType="beginValueType"/>
</xsd:simpleType>
<!-- mmtci:beginValue Type -->
<xsd:simpleType name="beginValueType">
  <xsd:union memberTypes="offsetType syncType eventType wallclockType"/>
</xsd:simpleType>
<!-- mmtci:end Type -->
<xsd:simpleType name="endType">
  <xsd:list itemType="endValueType"/>
</xsd:simpleType>
<!-- mmtci:endValue Type -->
<xsd:simpleType name="endValueType">
  <xsd:union memberTypes="offsetType syncType eventType wallclockType indefiniteType"/>
</xsd:simpleType>
<!-- mmtci:offsetType -->
<xsd:simpleType name="offsetType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-9]+?(\\. [0-9]+)?(h|min|s|ms)"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:durType -->
<xsd:simpleType name="durType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-9]+?(\\. [0-9]+)?(h|min|s|ms)"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:sync Type -->
<xsd:simpleType name="syncType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="([i-[:]][c-[:]]*\\.)?[i-[:]][c-[:]]*\\. (begin|end)"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:event Type -->
<xsd:simpleType name="eventType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="([i-[:]][c-[:]]*\\.)?[i-[:]][c-[:]]*\\. \c+"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:wallclockType -->
<xsd:simpleType name="wallclockType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-2][0-9]:[0-5][0-9](:[0-5][0-9](\\. [0-9]+)?)" />
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:indefiniteType -->
<xsd:simpleType name="indefiniteType">
  <xsd:restriction base="xsd:token">
```

```

        <xsd:enumeration value="indefinite"/>
    </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:clipBeginType -->
<xsd:simpleType name="clipBeginType">
    <xsd:union memberTypes="offsetType wallclockType MPUidType"/>
</xsd:simpleType>
<!-- mmtci:clipEndType -->
<xsd:simpleType name="clipEndType">
    <xsd:union memberTypes="offsetType wallclockType MPUidType"/>
</xsd:simpleType>
<!-- mmtci:MPUid Type -->
<xsd:simpleType name="MPUidType">
    <xsd:restriction base="xsd:token">
        <xsd:length value="256"/>
    </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:refDiv Type -->
<xsd:complexType name="refDivType">
    <xsd:simpleContent>
        <xsd:restriction base="xsd:ID"/>
    </xsd:simpleContent>
</xsd:complexType>
<!-- mmtci:viewRole Type -->
<xsd:simpleType name="viewRoleType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="default"/>
        <xsd:enumeration value="multiple"/>
        <xsd:enumeration value="receptible"/>
    </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:refId Type -->
<xsd:complexType name="refIdType">
    <xsd:simpleContent>
        <xsd:restriction base="xsd:ID"/>
    </xsd:simpleContent>
</xsd:complexType>
<!-- mmtci:mediaSrcType -->
<xsd:complexType name="mediaSrcType">
    <xsd:simpleContent>
        <xs:extension base="xsd:string">
            <xs:attribute name="mimeType" type="xs:string" use="required"/>
            <xsd:attribute name="isDependent" type="xsd:boolean" use="optional"
default="false"/>
            <xsd:attribute name="depId" type="StringVectorType" use="optional"/>
        </xs:extension>
    </xsd:simpleContent>
</xsd:complexType>
<!-- Type for space delimited list of strings -->
<xsd:simpleType name="StringVectorType">
    <xsd:list itemType="xsd:string"/>
</xsd:simpleType>

```

5.3.15 The *obsolete* attribute

The *obsolete* attribute indicates if the current update statement is obsolete and thus can safely be ignored by the user agent processing the current CI update. When not present, it is up to the user agent to find out using begin and end/dur attributes as well as subsequent update statements with the same reference.

5.3.16 The *obsoleteTime* attribute

The *obsoleteTime* attribute indicates when the current update statement becomes obsolete and thus can safely be ignored by the user agent processing the current CI update and playing media at a presentation time that is past the indicated *obsoleteTime*. If present, the user agent uses its current playback time and the CI timestamp to prune out all update statements that have an earlier *obsoleteTime*.

5.4 The elements of a CI document

5.4.1 The CI element

5.4.1.1 Semantics

The **CI** element is the root element of a **CI** document. It contains two elements: **view** and **MediaSync**.

5.4.1.2 Syntax

```
<xsd:element name="CI" = "CIType"/>
<!-- mmtci:CI Type -->
<xsd:complexType name="CIType">
  <xsd:sequence>
    <xsd:element name="view" type="viewType" minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element name="MediaSync" type="MediaSyncType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="version" type="xsd:integer">
  <xsd:attribute name="timestamp" type="xsd:dateTime">
</xsd:complexType>
```

5.4.1.3 Attributes

The **CI** element has the following attributes:

5.4.1.3.1 The *version* attribute

the version attribute provides a version number of the current CI update. The version number shall start with 0 for the very first CI that is assigned to the HTML5. Subsequent version number shall be incremented by one.

5.4.1.3.2 The *timestamp* attribute

the timestamp attribute provides the timestamp in presentation time of the current CI update. The timestamp indicates the timestamp of the earliest update statement that was added to the current CI update. All update statements that have a later or equal begin time are new update statements in the CI update.

5.4.2 The reference element

5.4.2.1 Semantics

The reference element provides a reference to the HTML5 document that the CI document updates. The reference shall contain a URI of the HTML5 document and may contain a version number. The version number, when present, shall match exactly one HTML5 document with the data-version attribute set to this version number.

5.4.2.2 Syntax

The reference element syntax is described in the following:

```
<xsd:element name="reference" type="ReferenceType"/>
<!-- mmtci:CI Type -->
<xsd:complexType name="ReferenceType">
  <xsd:complexContent>
    <xsd:extension base=xsd:anyURI>
      <xsd:attribute name="version" type="xsd:integer">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

5.4.3 The view element

5.4.3.1 Semantics

The **view** element is a child element of a **CI** element providing temporal information about layout changes of a View. The child elements of a **view** element describe Areas of which a View is composed of. This element may occur several times if many Views are to be presented.

5.4.3.2 Syntax

```
<!-- mmtci:view Type -->
<xsd:complexType name="viewType">
  <xsd:complexContent>
    <xsd:extension base="html5:HTMLElementType">
      <xsd:sequence>
        <xsd:element name="area" type="areaType" minOccurs="1"
maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="begin" type="beginType" default="0s"/>
      <xsd:attribute name="end" type="endType" default="indefinite"/>
      <xsd:attribute name="dur" type="durType"/>
      <xsd:attribute name="type" type="xsd:token" fixed="simple"/>
      <xsd:attribute name="href" type="xlink:hrefType"/>
      <xsd:attribute name="show" type="xsd:token" fixed="embed"/>
      <xsd:attribute name="actuate" type="xlink:actuateType"
default="onRequest"/>
      <xsd:attribute name="viewRole" type="viewRoleType" default="default"/>
      <xsd:attribute name="isLocked" type="xsd:boolean" use="optional"
default="true"/>
    </xsd:extension>
    <!-- The view element inherits HTMLElement interface which is specified in 3.2.2
of[W3C HTML5]-->
  </xsd:complexContent>
</xsd:complexType>
```

5.4.3.3 Attributes

The **view** element may have following attributes:

- *begin*: defined in [5.3.1](#) of this part of ISO/IEC 23008.
- *end*: defined in [5.3.2](#) of this part of ISO/IEC 23008.
- *dur*: defined in [5.3.3](#) of this part of ISO/IEC 23008.
- *xlink:type*: is fixed and have the value "simple".
- *xlink:href*: defined in [5.3.8](#) of this part of ISO/IEC 23008.
- *xlink:show*: is fixed and have the value "embed".
- *xlink:actuate*: defined in [5.3.9](#) of this part of ISO/IEC 23008.
- *viewRole*: defined in [5.3.10](#) of this part of ISO/IEC 23008.
- *isLocked*: When set to *true* it informs the rendering browser that the View referred to by **view** element cannot be modified. Therefore in order to allow personalization, at the time the view is created, the "isLocked" parameter shall be explicitly set to *false*. By choosing to have this new parameter at the view level it allows to either have the whole view or only some areas of the view personalizable.

5.4.4 The area element

5.4.4.1 Semantics

The **area** element is a child element of the **view** element. It provides spatial and temporal information of an Area. The spatial information is provided by the style attribute which is inherited from the HTMLElementType as defined in section Y. At the indicated begin time, the size of an Area can be increased or decreased, the Area may be shown or hidden, and any other style or spatial changes to the corresponding HTML5 element may be performed.

5.4.4.2 Syntax

```
<!-- mmtci:area Type -->
<xsd:complexType name="areaType">
  <xsd:complexContent>
    <xsd:extension base="html5:HTMLElementType">
      <xsd:attribute name="begin" type="beginType" default="0s"/>
      <xsd:attribute name="end" type="endType"/>
      <xsd:attribute name="dur" type="durType"/>
      <xsd:attribute name="refDiv" type="refDivType"/>
      <xsd:attribute name="style" type="xsd:string"/>

      <xsd:attribute name="isLocked" type="xsd:boolean" use="optional"
default="true"/>
      <xsd:attribute name="isLockedToView" type="xsd:boolean" use="optional"
default="true"/>
      <xsd:attribute name="minVisibility" type="Percentage" use="optional"
default="0"/>
    </xsd:extension>
    <!-- The view element inherits HTMLElement interface which is specified in 3.2.2
of[W3C HTML5]-->
  </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="Percentage">
  <xsd:restriction base="xsd:unsignedShort">
    <xsd:minInclusive value="0"/>
    <xsd:maxInclusive value="100"/>
  </xsd:restriction>
</xsd:simpleType>
```

5.4.4.3 Attributes

The **area** element includes following attributes:

- *begin*: defined in [5.3.1](#) of this part of ISO/IEC 23008.
- *end*: defined in [5.3.2](#) of this part of ISO/IEC 23008.
- *dur*: defined in [5.3.3](#) of this part of ISO/IEC 23008.
- *refDiv*: defined in [5.3.6](#) of this part of ISO/IEC 23008.
- *style*: defined in [5.3.7](#) of this part of ISO/IEC 23008.
- *isLocked*: When set to *true* it informs the rendering browser that the Area referred to by **area** elements cannot be modified. Therefore in order to allow personalization, at the time the view is created, the “isLocked” parameter shall be explicitly set to *false*. By choosing to have this new parameter at the area level it allows to either have the whole view or only some areas of the view personalizable.
- *isLockedToView*: When set to *true*, it informs the personalization system that the Area cannot be moved to another View.

- *minVisibility*: informs the rendering browser about how much visibility of the corresponding Area shall be always ensured. A “minVisibility” of “100” on an Area ensures that no other Area could overlap with it. The default value of this new parameter is “0” in order to be compatible with the default behavior of HTML5 and CSS. The visibility value of an Area is obtained by computing the ratio between the surface of the Area and the surface of all other Areas overlapping with the Area. In case, transparency is used, the surface of all Areas (main and overlapping ones) is weighted by the amount of transparency as shown in the equation below:

$$Visibility(Area_N) = \frac{Transparency(Area_N) * Surface(Area_N)}{\sum_{i=1}^{i=n} Transparency(Area_i) * Surface(Area_i)}$$

where

$Area_i$ refers to all Areas overlapping with $Area_N$.

5.4.5 The MediaSync element

5.4.5.1 Semantics

The **MediaSync** element is a child element of the **CI** element. It provides temporal information of the corresponding media elements of the HTML5 document. The CI document assigns media data (such as an MMT Asset, MPU, or a DASH MPD) to be played by a particular HTML5 media element at the indicated begin time.

5.4.5.2 Syntax

```
<!-- mmtci:MediaSync Type -->
<xsd:complexType name="MediaSyncType">
  <xsd:sequence>
    <xsd:element name="sourceList" type="sourceListType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="begin" type="beginType" default="0s"/>
  <xsd:attribute name="end" type="endType" default="indefinite"/>
  <xsd:attribute name="dur" type="durType"/>
  <xsd:attribute name="clipBegin" type="clipBeginType" default="0s"/>
  <xsd:attribute name="clipEnd" type="clipEndType" default="indefinite"/>
  <xsd:attribute name="refId" type="refIdType"/>
  <xsd:attribute name="type" type="xsd:token" fixed="simple"/>
  <xsd:attribute name="href" type="xlink:hrefType"/>
  <xsd:attribute name="actuate" type="xlink:actuateType"/>
</xsd:complexType>
<!-- mmtci:sourceList Type -->
<xsd:complexType name="sourceListType">
  <xsd:attribute name="mediaSrc" type="mediaSrcType"/> </xsd:complexType>
```

5.4.5.3 Attributes

The **MediaSync** element includes following attributes:

- *begin*: defined in [5.3.1](#) of this part of ISO/IEC 23008.
- *end*: defined in [5.3.2](#) of this part of ISO/IEC 23008.
- *dur*: defined in [5.3.3](#) of this part of ISO/IEC 23008.
- *clipBegin*: defined in [5.3.4](#) of this part of ISO/IEC 23008.
- *clipEnd*: defined in [5.3.5](#) of this part of ISO/IEC 23008.
- *refId*: defined in [5.3.12](#) of this part of ISO/IEC 23008.

- *xlink:type*: fixed “simple”.
- *xlink:href*: defined in 5.3.8 of this part of ISO/IEC 23008.
- *xlink:actuate*: defined in 5.3.9 of this part of ISO/IEC 23008.
- *mediaSrc*: defined in 5.3.13 of this part of ISO/IEC 23008.

5.4.6 The sourceList element

5.4.6.1 Semantics

The **sourceList** element is a child element of the **MediaSync** element. It provides the list of Assets to be presented in sequence.

5.4.6.2 Syntax

```
<!-- mmtci:sourceList Type -->
<xsd:complexType name="mmtci:sourceList">
  <xsd:sequence>
    <xsd:element name="mediaSrc" type="mmtci:mediaSrcType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

5.4.6.3 Attributes

The **sourceList** element includes following attributes:

- *mediaSrc*: defined in 5.3.13 of this part of ISO/IEC 23008.
- *contentType*: provides the MIME type of the media data
- *isDependent*: defined in 5.3.10 of this part of ISO/IEC 23008.
- *depAssets depId*: defined in 5.3.11 of this part of ISO/IEC 23008.

6 HTML data attributes

6.1 Introduction

This part of ISO/IEC 23008 defines a set of data attributes that may be added to the head element of the HTML5 document. These attributes convey information to facilitate mapping the HTML5 document and the corresponding CI documents.

6.2 Attributes

6.2.1 data-version attribute

The data-version attribute carries the version number of this document. The version number must be incremented by 1 for each new HTML5 document that has the same URI. This attribute is optional and is by default set to 1.

6.2.2 data-ci attribute

The data-ci attribute carries the URI of the CI document that will provide the updates to this HTML5 document. This attribute is optional.

7 Update Mechanism

7.1 Introduction

The Composition Information consists of the HTML5 document, which serves as the entry point to the presentation, as well as any accompanying file such as script files, style sheets, and the CI document. All of the latter ones are referenced from the HTML5 document. A CI document might be referenced from within a script that launches the CI parser.

In order to ensure smooth updates to the presentation, all update commands are confined to the CI document. This prevents complete document refresh as the DOM, created out of the HTML5, is modified gradually instead of being completely reloaded whenever an update becomes available. This is typically the case because loading a complete HTML5 document usually results in removing the old DOM and building a new DOM, which will likely impact the rendering of the presentation.

The presentation author shall create a new version of the CI whenever an update to the presentation is required. The CI shall be assigned a version number that is different and monotonically increasing. Each CI version shall be a superset of the previous CI version, i.e. it shall contain all update commands from the previous CI version. The client or user agent shall check the version number before processing the rest of the CI document. If the version number of the CI document is unchanged, the CI document does not need to be processed.

The CI shall contain a time indication that represents the timestamp of the first new update command that was added to the current CI version of the document, which shall be equal to the begin time of that update command. This timestamp indicates the latest time by which the current version of the CI obsoletes the previous version of the CI.

Whenever the client or user agent receives a new version of the CI, it shall first prune the update commands that are no longer applicable. An update command is no longer applicable if its end time has been reached or if it has been overwritten by a new update command. To facilitate the processing, the CI author may mark the instructions that are obsolete by that time. A client or user agent shall ignore update instructions that are obsolete by the actual presentation time.

Annex A (informative)

XML Schema of Composition Information

A.1 Syntax

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:html5="http://www.w3.
org/1999/xhtml" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ns1="mmtci"
targetNamespace="mmtci" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:element name="CI" type="CIType"/>
  <!-- mmtci:CI Type -->
  <xsd:complexType name="CIType">
    <xsd:sequence>
      <xsd:element name="view" type="viewType" minOccurs="1"
maxOccurs="unbounded"/>
      <xsd:element name="MediaSync" type="MediaSyncType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <!-- mmtci:view Type -->
  <xsd:complexType name="viewType">
    <xsd:complexContent>
      <xsd:extension base="html5:HTMLElementType">
        <xsd:sequence>
          <xsd:element name="area" type="areaType" minOccurs="1"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="begin" type="beginType" default="0s"/>
        <xsd:attribute name="end" type="endType" default="indefinite"/>
        <xsd:attribute name="dur" type="durType"/>
        <xsd:attribute name="type" type="xsd:token" fixed="simple"/>
        <xsd:attribute name="href" type="xlink:hrefType"/>
        <xsd:attribute name="show" type="xsd:token" fixed="embed"/>
        <xsd:attribute name="actuate" type="xlink:actuateType"
default="onRequest"/>
        <xsd:attribute name="viewRole" type="viewRoleType" default="default"/>
      </xsd:extension>
      <!-- The view element inherits HTMLElement interface which is specified in
3.2.2 of [W3C HTML5] -->
    </xsd:complexContent>
  </xsd:complexType>
  <!-- mmtci:area Type -->
  <xsd:complexType name="areaType">
    <xsd:complexContent>
      <xsd:extension base="html5:HTMLElementType">
        <xsd:attribute name="begin" type="beginType" default="0s"/>
        <xsd:attribute name="end" type="endType" default="indefinite"/>
        <xsd:attribute name="dur" type="durType"/>
        <xsd:attribute name="refDiv" type="refDivType"/>
      </xsd:extension>
      <!-- The view element inherits HTMLElement interface which is specified in
3.2.2 of [W3C HTML5] -->
    </xsd:complexContent>
  </xsd:complexType>

  <!-- mmtci:MediaSync Type -->
  <xsd:complexType name="MediaSyncType">
    <xsd:sequence>
      <xsd:element name="sourceList" type="sourceListType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>

```

```

<xsd:attribute name="begin" type="beginType" default="0s"/>
<xsd:attribute name="end" type="endType" default="indefinite"/>
<xsd:attribute name="dur" type="durType"/>
<xsd:attribute name="clipBegin" type="clipBeginType" default="0s"/>
<xsd:attribute name="clipEnd" type="clipEndType" default="indefinite"/>
<xsd:attribute name="refId" type="refIdType"/>
  <xsd:attribute name="type" type="xsd:token" fixed="simple"/>
<xsd:attribute name="href" type="xlink:hrefType"/>
<xsd:attribute name="show" type="xsd:token" fixed="embed"/>
<xsd:attribute name="actuate" type="xlink:actuateType"/>
<xsd:attribute name="isDependent" type="xsd:boolean" use="optional"
default="false"/>
  <xsd:attribute name="depId" type="StringVectorType" use="optional"/>
</xsd:complexType>
<!-- mmtci:sourceList Type -->
<xsd:complexType name="mediaSrcType">
  <xsd:simpleContent>
    <xs:extension base="xsd:string">
      <xsd:attribute name="mimeType" type="xs:string" use="required"/>
    </xs:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!-- mmtci:begin Type -->
<xsd:simpleType name="beginType">
  <xsd:list itemType="beginValueType"/>
</xsd:simpleType>
<!-- mmtci:beginValue Type -->
<xsd:simpleType name="beginValueType">
  <xsd:union memberTypes="offsetType syncType eventType wallclockType"/>
</xsd:simpleType>
<!-- mmtci:end Type -->
<xsd:simpleType name="endType">
  <xsd:list itemType="endValueType"/>
</xsd:simpleType>
<!-- mmtci:endValue Type -->
<xsd:simpleType name="endValueType">
  <xsd:union memberTypes="offsetType syncType eventType wallclockType
indefiniteType"/>
</xsd:simpleType>
<!-- mmtci:offsetType -->
<xsd:simpleType name="offsetType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-9]+?(\\. [0-9]+)?(h|min|s|ms)"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:durType -->
<xsd:simpleType name="durType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-9]+?(\\. [0-9]+)?(h|min|s|ms)"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:sync Type -->
<xsd:simpleType name="syncType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="([i-[:]] [c-[:]]*\\.)?[i-[:]] [c-[:]]*\\. (begin|end)"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:event Type -->
<xsd:simpleType name="eventType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="([i-[:]] [c-[:]]*\\.)?[i-[:]] [c-[:]]*\\. \.c+"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:wallclockType -->
<xsd:simpleType name="wallclockType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-2][0-9]:[0-5][0-9](:[0-5][0-9](\\. [0-9]+)?)" />
  </xsd:restriction>
</xsd:simpleType>
<!-- mmtci:indefiniteType -->
<xsd:simpleType name="indefiniteType">

```

```

        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="indefinite"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!-- mmtci:clipBeginType -->
    <xsd:simpleType name="clipBeginType">
        <xsd:union memberTypes="offsetType wallclockType MPUidType"/>
    </xsd:simpleType>
    <!-- mmtci:clipEndType -->
    <xsd:simpleType name="clipEndType">
        <xsd:union memberTypes="offsetType wallclockType MPUidType"/>
    </xsd:simpleType>
    <!-- mmtci:MPUid Type -->
    <xsd:simpleType name="MPUidType">
        <xsd:restriction base="xsd:token">
            <xsd:length value="256"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!-- mmtci:refDiv Type -->
    <xsd:complexType name="refDivType">
        <xsd:simpleContent>
            <xsd:restriction base="xsd:ID"/>
        </xsd:simpleContent>
    </xsd:complexType>
    <!-- mmtci:viewRole Type -->
    <xsd:simpleType name="viewRoleType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="multiple"/>
            <xsd:enumeration value="receptible"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!-- mmtci:refId Type -->
    <xsd:complexType name="refIdType">
        <xsd:simpleContent>
            <xsd:restriction base="xsd:ID"/>
        </xsd:simpleContent>
    </xsd:complexType>
    <!-- mmtci:mediaSrcType -->
    <xsd:complexType name="mediaSrcType">
        <xsd:simpleContent>
            <xs:extension base="xsd:string">
                <xs:attribute name="mimeType" type="xs:string" use="required"/>
                <xs:attribute name="isDependent" type="xsd:boolean" use="optional"
default="false"/>
                <xs:attribute name="depId" type="StringVectorType" use="optional"/>
            </xs:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <!-- Type for space delimited list of strings -->
    <xsd:simpleType name="StringVectorType">
        <xsd:list itemType="xsd:string"/>
    </xsd:simpleType>
</xsd:schema>

```

Annex B (informative)

Examples of CI document

B.1 Introduction

Various multimedia services can be provided using MMT. For each service, CI data can be formed differently. This Annex shows an example of how CI data can be described in a CI document for several use cases.

B.2 Use case 1: Area change

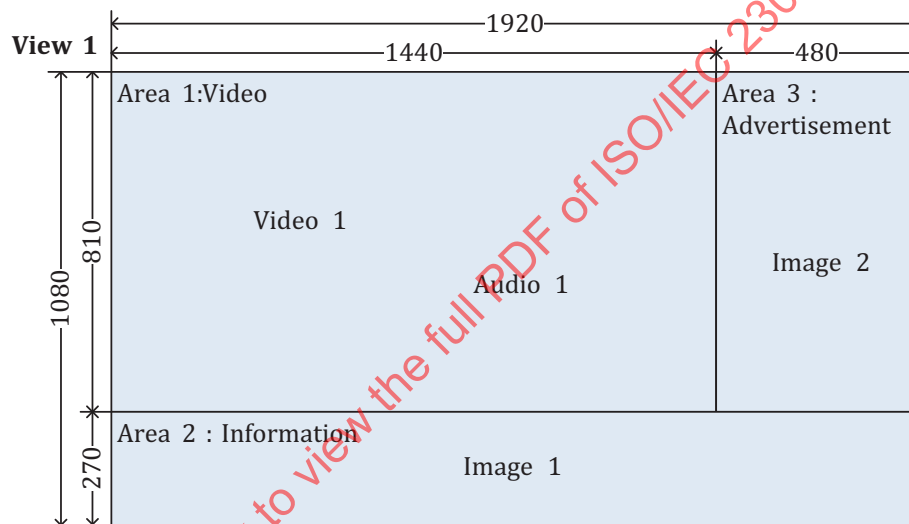


Figure B.1 — Initial View of Areas

Let assume that there are 7 different types of Asset in a Package. Initial as View is shown in [Figure B.1](#). The initial View then is updated as by follows:

Image1 in Area 2 is updated to Image3 at 18:00 (wall-clock time).

If Image2 in Area3 is selected, Area3 is splitted to Area4 and Area5 as shown in [Figure B.2](#) below. Area4 represents Widget 1 and Area 5 represents Image 4.

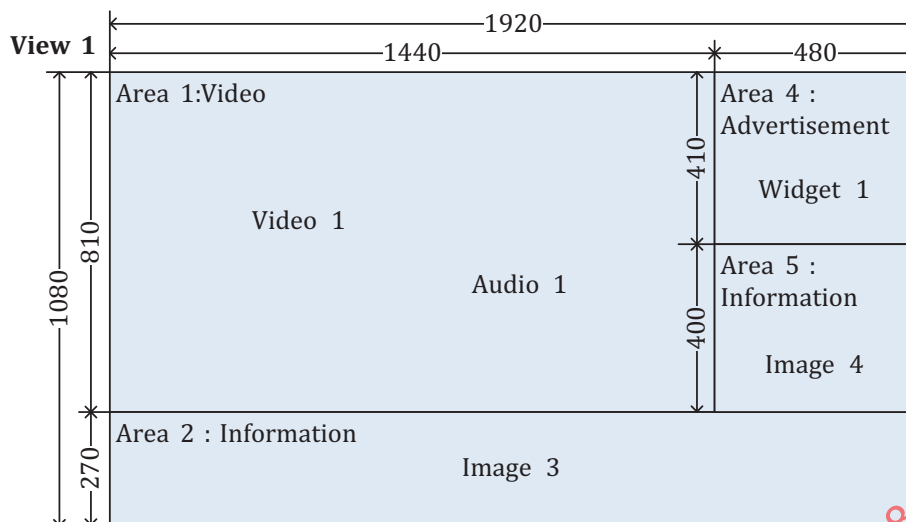


Figure B.2 — Later View of Areas after change

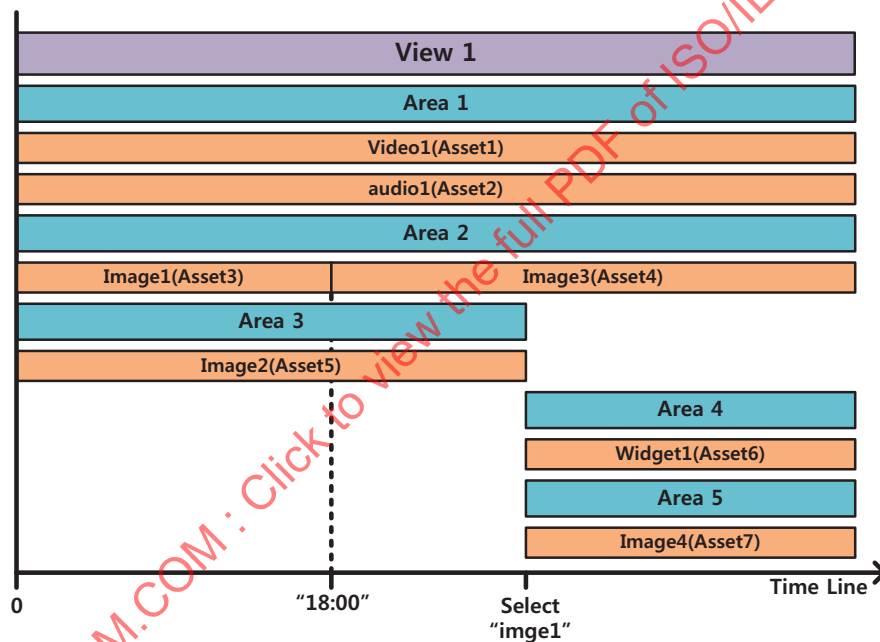


Figure B.3 — Timeline of Area change

— HTML5 document:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Entertainment</title>
  </head>
  <body>
    <div id="Area1" style="position:absolute; width:1440px; height:810px">
      <video id="Video1" src="asset1.m2v" style="position:absolute; left:0px;
top:0px; width:1440px; height:810px"/>
      <audio id="Audio1" src="asset2.mp3"/>
    </div>
    <div id="Area2" style="position:absolute; width:1920px; height:270px">
      
    </div>
  </body>
</html>
```

```

        
    </div>
    <div id="Area3" style="position:absolute; width:480px; height:810px">
        
    </div>
    <div id="Area4" style="position:absolute; width:480px; height:410px">
        <object id="Widget1" src="asset6.bt" style="position:absolute; left:0px;
top:0px; width:480px; height:410px"/>
    </div>
    <div id="Area5" style="position:absolute; width:480px; height:400px">
        
    </div>
</body>
</html>

```

— CI document:

```

<mmtci:CI>
    <mmtci:view id="View1" style="position:absolute; width:1920px; height:1080px"
mmtci:begin="0s" mmtci:end="indefinite">
        <mmtci:area id="divL1" style="position:absolute; left:0px; top:0px;
width:1440px; height:810px" mmtci:begin="0s" mmtci:end="indefinite" mmtci:refDiv="Area1"/>
        <mmtci:area id="divL2" style="position:absolute; left:0px; top:810px;
width:1920px; height:270px" mmtci:begin="0s" mmtci:end="indefinite" mmtci:refDiv="Area2"/>
        <mmtci:area id="divL3" style="position:absolute; left:1440px;
top:0px; width:480px; height:810px" mmtci:begin="0s" mmtci:end="image2.Click"
mmtci:refDiv="Area3"/>
        <mmtci:area id="divL4" style="position:absolute; left:1440px; top:0px;
width:480px; height:410px" mmtci:begin="image2.Click" mmtci:end="indefinite"
mmtci:refDiv="Area4"/>
        <mmtci:area id="divL5" style="position:absolute; left:1440px; top:410px;
width:480px; height:400px" mmtci:begin="image2.Click" mmtci:end="indefinite"
mmtci:refDiv="Area5"/>
    </mmtci:view>
    <mmtci:MediaSync refId="Video1" mmtci:begin="0s"/>
    <mmtci:MediaSync refId="Audio1" mmtci:begin="0s"/>
    <mmtci:MediaSync refId="Image1" mmtci:begin="0s" mmtci:end="18:00"/>
    <mmtci:MediaSync refId="Image2" mmtci:begin="0s"/>
    <mmtci:MediaSync refId="Image3" mmtci:begin="Image1.end"/>
    <mmtci:MediaSync refId="Widget1" mmtci:begin="0s"/>
    <mmtci:MediaSync refId="Image4" mmtci:begin="0s"/>
</mmtci:CI>

```

B.3 Use case 2: View change

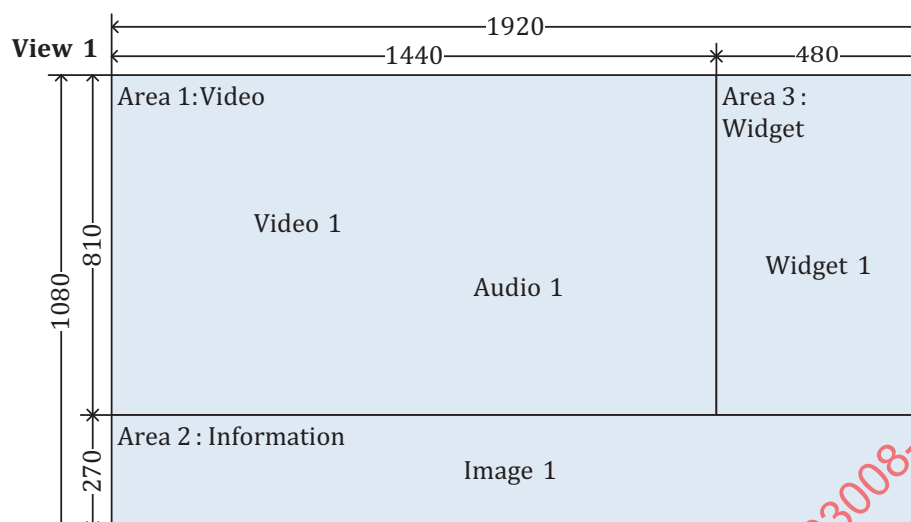


Figure B.4 — Initial View of for use case 2

Let assume that there are 2 Views and 7 different types of Asset in a Package and the initial View is shown in [Figure B.4](#). The initial View then is changed as by follows:

If Image1 in Area2 is selected, View 1 is update to View 2 as shown in [Figure B.5](#) below.

After updating, Vidio1 and Audio1 are played continuously. And Video2 in Area5 is in-sync with Video1. In this example, Video2 will start at the same time with Video1. However, video2 does not display. It will be activated with its Area (Area4) when clicked and displayed on the screen.

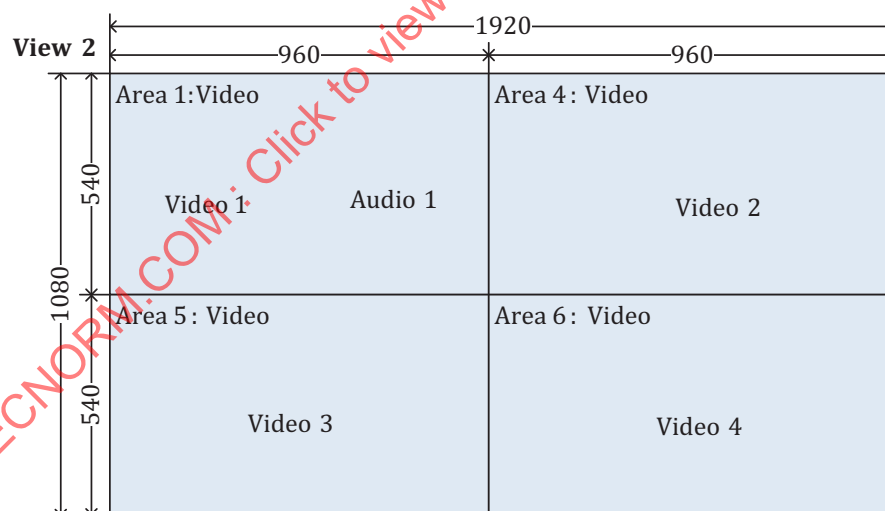


Figure B.5 — Later View of View change

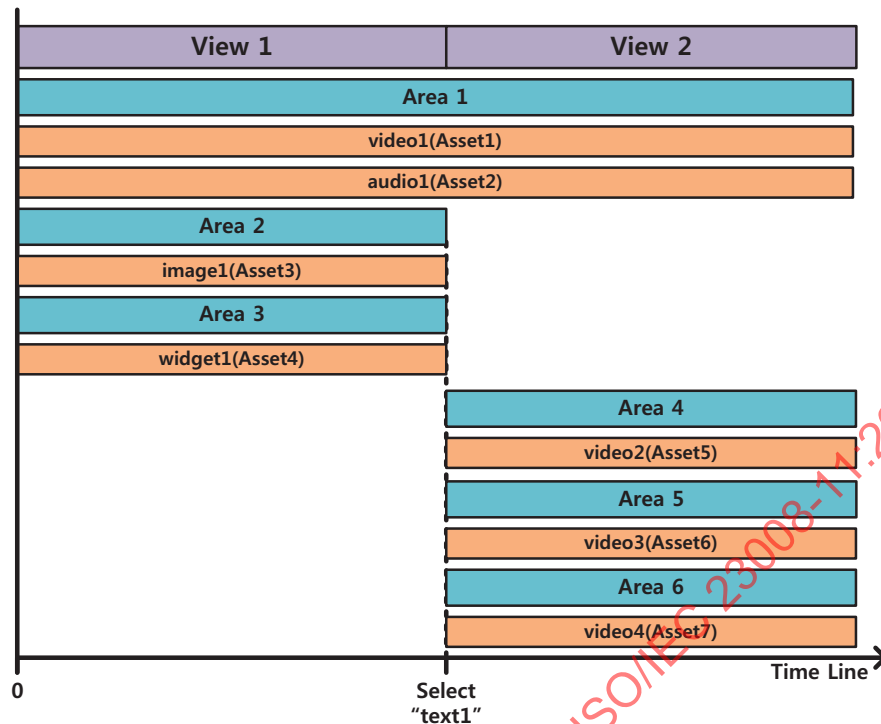


Figure B.6 — Timeline of View change

— HTML5 document:

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Sports</title>
  </head>
  <body>
    <div id="Area1" style="position:absolute; width:1440px; height:810px">
      <video id="video1" src="asset1.m2v" style="position:absolute; left:0px;
top:0px; width:1440px; height:810px"/>
      <audio id="audio1" src="asset2.mp3" mmctci:begin="0s"/>
    </div>
    <div id="Area2" style="position:absolute; width:1920px; height:270px">
      
    </div>
    <div id="Area3" style="position:absolute; width:480px; height:810px">
      <object id="widget1" src="asset4.bt" style="position:absolute; left:0px;
top:0px; width:480px; height:810px"/>
    </div>
    <div id="Area4" style="position:absolute; width:960px; height:540px">
      <video id="video2" src="asset5.m2v" style="position:absolute; left:0px;
top:0px; width:960px; height:540px"/>
    </div>
    <div id="Area5" style="position:absolute; width:960px; height:540px">
      <video id="video3" src="asset6.m2v" style="position:absolute; left:0px;
top:0px; width:960px; height:540px"/>
    </div>
    <div id="Area6" style="position:absolute; width:960px; height:540px">
      <video id="video4" src="asset7.m2v" style="position:absolute; left:0px;
top:0px; width:960px; height:540px"/>
    </div>
  </body>
</html>

```

— CI document:

```
<mmtci:CI>
  <mmtci:view id="View1" style="position:absolute; width:1920px; height:1080px"
mmtci:begin="0s" mmtci:end="text1.Click">
  <mmtci:area id="divL1" style="position:absolute; left:0px; top:0px; width:1440px;
height:810px" mmtci:begin="0s" mmtci:end="indefinite" mmtci:refDiv="Area1"/>
  <mmtci:area id="divL2" style="position:absolute; left:0px; top:810px; width:1920px;
height:270px" mmtci:begin="0s" mmtci:end="indefinite" mmtci:refDiv="Area2"/>
  <mmtci:area id="divL3" style="position:absolute; left:1440px; top:0px; width:480px;
height:810px" mmtci:begin="0s" mmtci:end="indefinite" mmtci:refDiv="Area3"/>
</mmtci:view>
  <mmtci:view id="View2" style="position:absolute; width:1920px; height:1080px"
mmtci:begin="text1.Click">
  <mmtci:area id="divL4" style="position:absolute; left:0px; top:0px; width:960px;
height:540px" mmtci:begin="0s" mmtci:end="indefinite" mmtci:refDiv="Area1"/>
  <mmtci:area id="divL5" style="position:absolute; left:960px; top:0px; width:960px;
height:540px" mmtci:begin="0s" mmtci:end="indefinite" mmtci:refDiv="Area4"/>
  <mmtci:area id="divL6" style="position:absolute; left:0px; top:540px; width:960px;
height:540px" mmtci:begin="0s" mmtci:end="indefinite" mmtci:refDiv="Area5"/>
  <mmtci:area id="divL7" style="position:absolute; left:960px; top:540px; width:960px;
height:540px" mmtci:begin="0s" mmtci:end="indefinite" mmtci:refDiv="Area6"/>
</mmtci:view>
  <mmtci:MediaSync refId="Video1" mmtci:begin="0s"/>
  <mmtci:MediaSync refId="audio1" mmtci:begin="0s"/>
  <mmtci:MediaSync refId="Image1" mmtci:begin="0s"/>
  <mmtci:MediaSync refId="widget1" mmtci:begin="0s"/>
  <mmtci:MediaSync refId="video2" mmtci:begin="Video1.begin"/>
  <mmtci:MediaSync refId="video3" mmtci:begin="0s"/>
  <mmtci:MediaSync refId="video4" mmtci:begin="0s"/>
</mmtci:CI>
```

B.4 Use case 3: Multi-screen presentation – Asset sharing

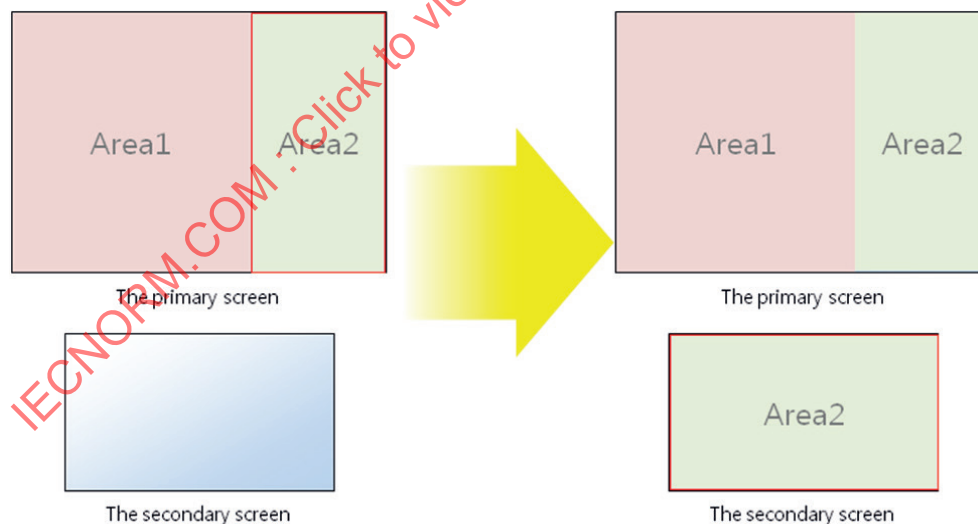


Figure B.7 — Presentation of Asset sharing

This use case is a simple example of Asset sharing. In this use case, an Area or Areas in a primary screen can be shared with the secondary screen after beginning of multi-screen presentation. [Figure B.7](#) shows how the presentation changes in this use case.

— HTML5 document:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Entertainment</title>
</head>
<body>
  <div id="Area1" style="position:absolute; width:1000px; height:1000px">
    <video id="video1" src="package1/asset1" style="position:absolute; width:100%;
height:100%; left:0px; top:0px"/>
  </div>
  <div id="Area2" style="position:absolute; width:600px; height:1000px">
    <video id="video2" src="package1/asset2" style="position:absolute; width:100%;
height:100%; left:0px; top:0px"/>
  </div>
</body>
</html>
```

— CI document:

```
<mmtci:CI>
  <mmtci:view id="View1" mmtci:viewRole="default" style="position:absolute;
width:1920px; height:1080px">
    <mmtci:area id="divL1" style="position:absolute; width:70%; height:100%;
left:0%; top:0%" mmtci:refDiv="Area1"/>
    <mmtci:area id="divL2" style="position:absolute; width:30%; height:100%;
left:70%; top:0%" mmtci:refDiv="Area2"/>
  </mmtci:view>
  <mmtci:view id="View2" mmtci:viewRole="multiple" style="position:absolute;
width:1920px; height:1080px">
    <mmtci:area id="divL1" style="position:absolute; width:70%; height:100%;
left:0%; top:0%" mmtci:refDiv="Area1"/>
    <mmtci:area id="divL2" style="position:absolute; width:30%; height:100%;
left:70%; top:0%" mmtci:refDiv="Area2"/>
  </mmtci:view>
  <mmtci:MediaSync refId="video1" mmtci:begin="0s"/>
  <mmtci:MediaSync refId="video2" mmtci:begin="0s"/>
</mmtci:CI>
```

B.5 Use case 4: Multi-screen presentation – dynamic Asset sharing

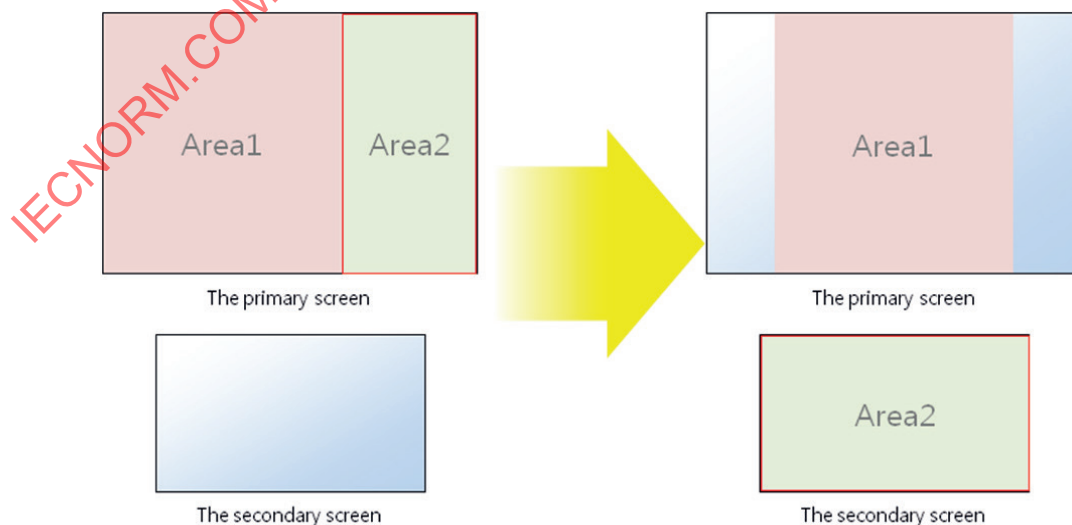


Figure B.8 — Presentation of dynamic Asset sharing

This use case is a simple example of Dynamic Asset Sharing. In this use case, one of the Areas in a primary screen moves to the secondary screen after the beginning of multi-screen presentation, and the one Area remained in the primary screen is repositioned in the center, as shown in [Figure B.8](#) above.

— HTML5 document

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Entertainment</title>
</head>
<body>
  <div id="Area1" style="position:absolute; width:1000px; height:1000px">
    <video id="video1" src="package1/asset1" style="position:absolute; width:100%;
height:100%; left:0px; top:0px"/>
  </div>
  <div id="Area2" style="position:absolute; width:600px; height:1000px">
    <video id="video2" src="package1/asset2" style="position:absolute; width:100%;
height:100%; left:0px; top:0px"/>
  </div>
</body>
</html>
```

— CI document:

```
<mmtci:CI>
  <mmtci:view id="View1" mmtci:viewRole="default" style="position:absolute;
width:1920px; height:1080px">
    <mmtci:area id="divL1" style="position:absolute; width:70%; height:100%;
left:0%; top:0%" mmtci:refDiv="Area1"/>
    <mmtci:area id="divL2" style="position:absolute; width:30%; height:100%;
left:70%; top:0%" mmtci:refDiv="Area2"/>
  </mmtci:view>
  <mmtci:view id="View2" mmtci:viewRole="multiple" style="position:absolute;
width:1920px; height:1080px">
    <mmtci:area id="divL1" style="position:absolute; width:70%; height:100%;
left:0%; top:0%" mmtci:refDiv="Area1"/>
    <mmtci:area id="divL2" style="position:absolute; width:30%; height:100%;
left:70%; top:0%" mmtci:refDiv="Area2" mmtci:plungeOut="dynamic"/>
  </mmtci:view>
  <mmtci:MediaSync refId="video1" mmtci:begin="0s"/>
  <mmtci:MediaSync refId="video2" mmtci:begin="0s"/>
</mmtci:CI>
```

B.6 Use case 5: Multi-screen presentation – complementary Asset

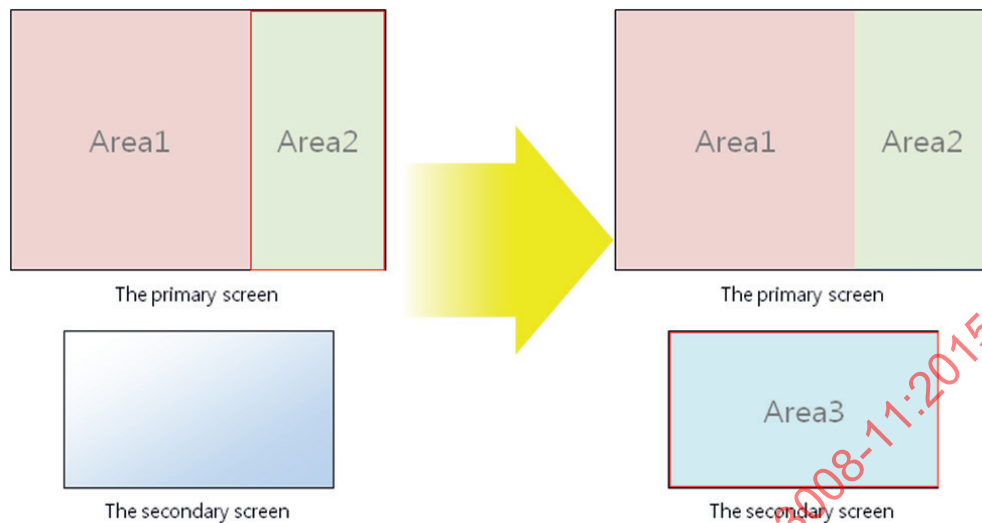


Figure B.9 — Presentation of complementary Asset

This use case is a simple example of complementary Asset presentation. In this use case an Area invisible in the primary screen moves to a secondary screen after the beginning of multi-screen presentation, as shown in [Figure B.9](#) above.

— HTML5 document:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Entertainment</title>
</head>
<body>
  <div id="Area1" style="position:absolute; width:1000px; height:1000px">
    <video id="video1" src="package1/asset1" style="position:absolute; width:100%;
height:100%; left:0px; top:0px"/>
  </div>
  <div id="Area2" style="position:absolute; width:600px; height:1000px">
    <video id="video2" src="package1/asset2" style="position:absolute; width:100%;
height:100%; left:0px; top:0px"/>
  </div>
  <div id="Area3" style="position:absolute; width:1024px; height:768px">
    <object id="widget1" src="package1/asset3" style="position:absolute; width:100%;
height:100%; left:0px; top:0px"/>
  </div>
</body>
</html>
```

— CI document:

```
<mmtci:CI>
  <mmtci:view id="View1" mmtci:viewRole="default" style="position:absolute;
width:1920px; height:1080px">
    <mmtci:area id="divL1" style="position:absolute; width:70%; height:100%;
left:0%; top:0%" mmtci:refDiv="Area1"/>
    <mmtci:area id="divL2" style="position:absolute; width:30%; height:100%;
left:70%; top:0%" mmtci:refDiv="Area2"/>
  </mmtci:view>
  <mmtci:view id="View2" mmtci:viewRole="multiple" style="position:absolute;
width:1920px; height:1080px">
    <mmtci:area id="divL1" style="position:absolute; width:70%; height:100%;
left:0%; top:0%" mmtci:refDiv="Area1"/>
    <mmtci:area id="divL2" style="position:absolute; width:30%; height:100%;
```

```

left:70%; top:0%" mmtci:refDiv="Area2"/>
  <mmtci:area id="divL3" mmtci:refDiv="Area3"/>
</mmtci:view>
<mmtci:MediaSync refId="video1" mmtci:begin="0s"/>
<mmtci:MediaSync refId="video2" mmtci:begin="0s"/>
<mmtci:MediaSync refId="widget1" mmtci:begin="0s"/>
</mmtci:CI >

```

B.7 Use case 6: Multi-screen presentation – Asset-level sharing

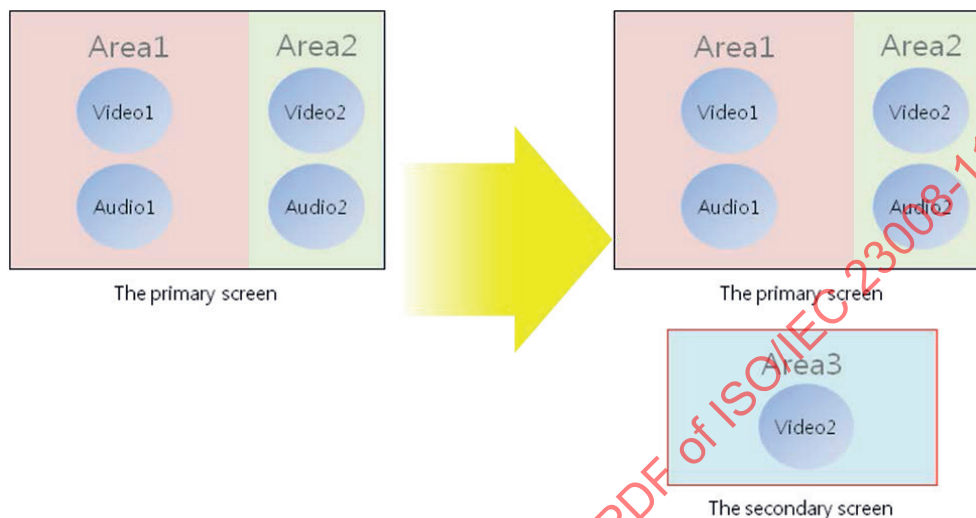


Figure B.10 — Asset-level sharing

This use case is a simple example of Asset-level sharing. In this use case, Area1 has audio1 and video1, and Area2 has audio2 and video2. Area3 has video2 only.

— HTML5 document:

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Entertainment</title>
</head>
<body>
  <div id="Area1" style="position:absolute; width:1000px; height:1000px">
    <video id="video1" src=" asset1"/>
    <audio id="Audio1" src=" asset2"/>
  </div>
  <div id="Area2" style="position:absolute; width:600px; height:1000px">
    <video id="video2" src=" asset3"/>
  </div>
  <div id="Area3">
    <video id="video2" src=" asset3"/>
  </div>
</body>
</html>

```

— CI document:

```

<mmtci:CI>
  <mmtci:view id="View1" mmtci:viewRole="default" style="position:absolute;
width:1920px; height:1080px">
    <mmtci:area id="divL1" style="position:absolute; width:70%; height:100%;
left:0%; top:0%" mmtci:refDiv="Area1"/>
    <mmtci:area id="divL2" style="position:absolute; width:30%; height:100%;

```

```

left:70%; top:0%" mmtci:refDiv="Area2"/>
</mmtci:view>
<mmtci:view id="View2" mmtci:viewRole="multiple" style="position:absolute;
width:1920px; height:1080px">
  <mmtci:area id="divL1" style="position:absolute; width:70%; height:100%;
left:0%; top:0%" mmtci:refDiv="Area1"/>
  <mmtci:area id="divL2" style="position:absolute; width:30%; height:100%;
left:70%; top:0%" mmtci:refDiv="Area2"/>
  <mmtci:area id="divL3" mmtci:refDiv="Area3"/>
</mmtci:view>
<mmtci:MediaSync refId="video1" mmtci:begin="0s"/>
<mmtci:MediaSync refId="Audio1" mmtci:begin="0s"/>
<mmtci:MediaSync refId="video2" mmtci:begin="0s"/>
<mmtci:MediaSync refId="Audio2" mmtci:begin="0s"/>
</mmtci:CI >

```

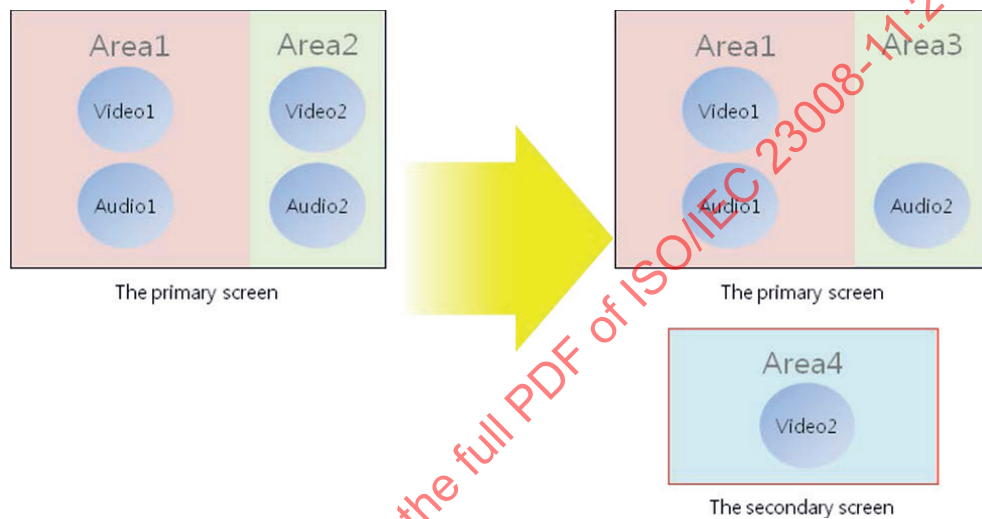


Figure B.11 — Another example of Asset-level sharing

Following is another example of Asset-level sharing. In this example, the primary device has only audio2, and video2 played in the secondary device.

— HTML5 document:

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Entertainment</title>
</head>
<body>
  <div id="Area1" style="position:absolute; width:1000px; height:1000px">
    <video id="video1" src=" asset1"/>
    <audio id="Audio1" src=" asset2"/>
  </div>
  <div id="Area2" style="position:absolute; width:600px; height:1000px">
    <video id="video2" src=" asset3"/>
  </div>
  <div id="Area3" style="position:absolute; width:600px; height:1000px">
    <audio id="Audio2" src=" asset4"/>
  </div>
  <div id="Area4">
    <video id="video2" src=" asset3"/>
  </div>
</body>
</html>

```