

```

initial-value [105] INTEGER OPTIONAL,
initial-portion [106] INTEGER OPTIONAL,
step-size [107] INTEGER DEFAULT 1,
slider-style [108] SliderStyle DEFAULT normal,
slider-ref-colour [50] Colour OPTIONAL
}

```

Orientation ::= ENUMERATED {left(1), right(2), up(3), down(4)}

SliderStyle ::= ENUMERATED {normal(1), thermometer(2), proportional(3)}

-- A.36 Entry Field Class _____

```

EntryFieldClass ::= SET
{
  COMPONENTS OF TextClass,
  COMPONENTS OF InteractableClass,
  input-type [109] InputType DEFAULT any,
  char-list [110] OCTET STRING OPTIONAL,
  obscured-input [111] BOOLEAN DEFAULT FALSE,
  max-length [112] INTEGER DEFAULT 0
}

```

InputType ::= ENUMERATED {alpha(1), numeric(2), any(3), listed(4)}

-- A.37 Hyper Text Class _____

```

HyperTextClass ::= SET
{
  COMPONENTS OF TextClass,
  COMPONENTS OF InteractableClass
}

```

-- A.38 Button Class _____

```

ButtonClass ::= SET
{
  COMPONENTS OF VisibleClass
  (WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT}),
  COMPONENTS OF InteractableClass,
  button-ref-colour [48] Colour OPTIONAL
}

```

-- A.39 Hotspot Class _____

HotspotClass ::= ButtonClass

-- A.40 Push Button Class _____

```
PushButtonClass ::= SET
{
  COMPONENTS OF ButtonClass,
  original-label [113] OCTET STRING OPTIONAL,
  character-set [38] INTEGER OPTIONAL
}
```

-- A.41 Switch Button Class _____

```
SwitchButtonClass ::= SET
{
  COMPONENTS OF PushButtonClass,
  button-style [114] ButtonStyle
}
```

```
ButtonStyle ::= ENUMERATED
{
  pushbutton(1),
  radiobutton(2),
  checkbox(3)
}
```

-- A.42 Action Class _____

ActionClass ::= SEQUENCE SIZE (1..MAX) OF ElementaryAction

```
ElementaryAction ::= CHOICE
{
  activate [115] GenericObjectReference,
  add [116] Add,
  add-item [117] AddItem,
  append [118] Append,
  bring-to-front [119] GenericObjectReference,
  call [120] Call,
  call-action-slot [121] CallActionSlot,
  clear [122] GenericObjectReference,
  clone [123] Clone,
  close-connection [124] CloseConnection,
  deactivate [125] GenericObjectReference,
```

del-item [126] DelItem,
 deselect [127] GenericObjectReference,
 deselect-item [128] DeselectItem,
 divide [129] Divide,
 draw-arc [130] DrawArc,
 draw-line [131] DrawLine,
 draw-oval [132] DrawOval,
 draw-polygon [133] DrawPolygon,
 draw-polyline [134] DrawPolyline,
 draw-rectangle [135] DrawRectangle,
 draw-sector [136] DrawSector,
 fork [137] Fork,
 get-availability-status [138] GetAvailabilityStatus,
 get-box-size [139] GetBoxSize,
 get-cell-item [140] GetCellItem,
 get-cursor-position [141] GetCursorPosition,
 get-engine-support [142] GetEngineSupport,
 get-entry-point [143] GetEntryPoint,
 get-fill-colour [144] GetFillColor,
 get-first-item [145] GetFirstItem,
 get-highlight-status [146] GetHighlightStatus,
 get-interaction-status [147] GetInteractionStatus,
 get-item-status [148] GetItemStatus,
 get-label [149] GetLabel,
 get-last-anchor-fired [150] GetLastAnchorFired,
 get-line-colour [151] GetLineColour,
 get-line-style [152] GetLineStyle,
 get-line-width [153] GetLineWidth,
 get-list-item [154] GetListItem,
 get-list-size [155] GetListSize,
 get-overwrite-mode [156] GetOverwriteMode,
 get-portion [157] GetPortion,
 get-position [158] GetPosition,
 get-running-status [159] GetRunningStatus,
 get-selection-status [160] GetSelectionStatus,
 get-slider-value [161] GetSliderValue,
 get-text-content [162] GetTextContent,
 get-text-data [163] GetTextData,
 get-token-position [164] GetTokenPosition,
 get-volume [165] GetVolume,
 launch [166] GenericObjectReference,
 lock-screen [167] GenericObjectReference,
 modulo [168] Modulo,
 move [169] Move,

move-to [170] MoveTo,
 multiply [171] Multiply,
 open-connection [172] OpenConnection,
 preload [173] GenericObjectReference,
 put-before [174] PutBefore,
 put-behind [175] PutBehind,
 quit [176] GenericObjectReference,
 read-persistent [177] ReadPersistent,
 run [178] GenericObjectReference,
 scale-bitmap [179] ScaleBitmap,
 scale-video [180] ScaleVideo,
 scroll-items [181] ScrollItems,
 select [182] GenericObjectReference,
 select-item [183] SelectItem,
 send-event [184] SendEvent,
 send-to-back [185] GenericObjectReference,
 set-box-size [186] SetBoxSize,
 set-cache-priority [187] SetCachePriority,
 set-counter-end-position [188] SetCounterEndPosition,
 set-counter-position [189] SetCounterPosition,
 set-counter-trigger [190] SetCounterTrigger,
 set-cursor-position [191] SetCursorPosition,
 set-cursor-shape [192] SetCursorShape,
 set-data [193] SetData,
 set-entry-point [194] SetEntryPoint,
 set-fill-colour [195] SetFillColor,
 set-first-item [196] SetFirstItem,
 set-font-ref [197] SetFontRef,
 set-highlight-status [198] SetHighlightStatus,
 set-interaction-status [199] SetInteractionStatus,
 set-label [200] SetLabel,
 set-line-colour [201] SetLineColor,
 set-line-style [202] SetLineStyle,
 set-line-width [203] SetLineWidth,
 set-overwrite-mode [204] SetOverwriteMode,
 set-palette-ref [205] SetPaletteRef,
 set-portion [206] SetPortion,
 set-position [207] SetPosition,
 set-slider-value [208] SetSliderValue,
 set-speed [209] SetSpeed,
 set-timer [210] SetTimer,
 set-transparency [211] SetTransparency,
 set-variable [212] SetVariable,
 set-volume [213] SetVolume,

```

spawn [214] GenericObjectReference,
step [215] Step,
stop [216] GenericObjectReference,
store-persistent [217] StorePersistent,
subtract [218] Subtract,
test-variable [219] TestVariable,
toggle [220] GenericObjectReference,
toggle-item [221] ToggleItem,
transition-to [222] TransitionTo,
unload [223] GenericObjectReference,
unlock-screen [224] GenericObjectReference
}

```

Add ::= SEQUENCE

```

{
    target GenericObjectReference,
    value GenericInteger
}

```

AddItem ::= SEQUENCE

```

{
    target GenericObjectReference,
    item-index GenericInteger,
    visible-reference GenericObjectReference
}

```

Append ::= SEQUENCE

```

{
    target GenericObjectReference,
    append-value GenericOctetString
}

```

Call ::= SEQUENCE

```

{
    target GenericObjectReference,
    call-succeeded ObjectReference,
    parameters SEQUENCE SIZE (1..MAX) OF Parameter OPTIONAL
}

```

CallActionSlot ::= SEQUENCE

```

{
    target GenericObjectReference,
    index GenericInteger
}

```

Clone ::= SEQUENCE

```
{
  target GenericObjectReference,
  clone-ref-var ObjectReference
}
```

CloseConnection ::= SEQUENCE

```
{
  target GenericObjectReference,
  connection-tag GenericInteger
}
```

DelItem ::= SEQUENCE

```
{
  target GenericObjectReference,
  visible-reference GenericObjectReference
}
```

DeselectItem ::= SEQUENCE

```
{
  target GenericObjectReference,
  item-index GenericInteger
}
```

Divide ::= SEQUENCE

```
{
  target GenericObjectReference,
  value GenericInteger
}
```

DrawArc ::= SEQUENCE

```
{
  target GenericObjectReference,
  x GenericInteger,
  y GenericInteger,
  ellipse-width GenericInteger,
  ellipse-height GenericInteger,
  start-angle GenericInteger,
  arc-angle GenericInteger
}
```

DrawLine ::= SEQUENCE

```
{
```

```
target GenericObjectReference,  
x1 GenericInteger,  
y1 GenericInteger,  
x2 GenericInteger,  
y2 GenericInteger  
}
```

```
DrawOval ::= SEQUENCE  
{  
    target GenericObjectReference,  
    x GenericInteger,  
    y GenericInteger,  
    ellipse-width GenericInteger,  
    ellipse-height GenericInteger  
}
```

```
DrawPolygon ::= SEQUENCE  
{  
    target GenericObjectReference,  
    pointlist SEQUENCE SIZE (1..MAX) OF Point  
}
```

```
DrawPolyline ::= SEQUENCE  
{  
    target GenericObjectReference,  
    pointlist SEQUENCE SIZE (1..MAX) OF Point  
}
```

```
DrawRectangle ::= SEQUENCE  
{  
    target GenericObjectReference,  
    x1 GenericInteger,  
    y1 GenericInteger,  
    x2 GenericInteger,  
    y2 GenericInteger  
}
```

```
DrawSector ::= SEQUENCE  
{  
    target GenericObjectReference,  
    x GenericInteger,  
    y GenericInteger,  
    ellipse-width GenericInteger,  
    ellipse-height GenericInteger,  
}
```

```
    start-angle GenericInteger,  
    arc-angle GenericInteger  
}
```

```
Fork ::= SEQUENCE  
{  
    target GenericObjectReference,  
    fork-succeeded ObjectReference,  
    parameters SEQUENCE SIZE (1..MAX) OF Parameter OPTIONAL  
}
```

```
GetAvailabilityStatus ::= SEQUENCE  
{  
    target GenericObjectReference,  
    availability-status-var ObjectReference  
}
```

```
GetBoxSize ::= SEQUENCE  
{  
    target GenericObjectReference,  
    x-box-size-var ObjectReference,  
    y-box-size-var ObjectReference  
}
```

```
GetCellItem ::= SEQUENCE  
{  
    target GenericObjectReference,  
    cell-index GenericInteger,  
    item-ref-var ObjectReference  
}
```

```
GetCursorPosition ::= SEQUENCE  
{  
    target GenericObjectReference,  
    x-out ObjectReference,  
    y-out ObjectReference  
}
```

```
GetEngineSupport ::= SEQUENCE  
{  
    target GenericObjectReference,  
    feature GenericOctetString,  
    answer ObjectReference  
}
```

```
GetEntryPoint ::= SEQUENCE
{
    target GenericObjectReference,
    entry-point-var ObjectReference
}
```

```
GetFillColour ::= SEQUENCE
{
    target GenericObjectReference,
    fill-colour-var ObjectReference
}
```

```
GetFirstItem ::= SEQUENCE
{
    target GenericObjectReference,
    first-item-var ObjectReference
}
```

```
GetHighlightStatus ::= SEQUENCE
{
    target GenericObjectReference,
    highlight-status-var ObjectReference
}
```

```
GetInteractionStatus ::= SEQUENCE
{
    target GenericObjectReference,
    interaction-status-var ObjectReference
}
```

```
GetItemStatus ::= SEQUENCE
{
    target GenericObjectReference,
    item-index GenericInteger,
    item-status-var ObjectReference
}
```

```
GetLabel ::= SEQUENCE
{
    target GenericObjectReference,
    label-var ObjectReference
}
```

GetLastAnchorFired ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    last-anchor-fired-var ObjectReference  
}
```

GetLineColour ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    line-colour-var ObjectReference  
}
```

GetLineStyle ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    line-style-var ObjectReference  
}
```

GetLineWidth ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    line-width-var ObjectReference  
}
```

GetListItem ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    item-index GenericInteger,  
    item-ref-var ObjectReference  
}
```

GetListSize ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    size-var ObjectReference  
}
```

GetOverwriteMode ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    overwrite-mode-var ObjectReference  
}
```

GetPortion ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    portion-var ObjectReference  
}
```

GetPosition ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    x-position-var ObjectReference,  
    y-position-var ObjectReference  
}
```

GetRunningStatus ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    running-status-var ObjectReference  
}
```

GetSelectionStatus ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    selection-status-var ObjectReference  
}
```

GetSliderValue ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    slider-value-var ObjectReference  
}
```

GetTextContent ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    text-content-var ObjectReference  
}
```

GetTextData ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    text-data-var ObjectReference  
}
```

GetTokenPosition ::= SEQUENCE

```
{
```

200

```
target GenericObjectReference,  
token-position-var ObjectReference  
}
```

```
GetVolume ::= SEQUENCE  
{  
target GenericObjectReference,  
volume-var ObjectReference  
}
```

```
Modulo ::= SEQUENCE  
{  
target GenericObjectReference,  
value GenericInteger  
}
```

```
Move ::= SEQUENCE  
{  
target GenericObjectReference,  
movement-identifier GenericInteger  
}
```

```
MoveTo ::= SEQUENCE  
{  
target GenericObjectReference,  
index GenericInteger  
}
```

```
Multiply ::= SEQUENCE  
{  
target GenericObjectReference,  
value GenericInteger  
}
```

```
OpenConnection ::= SEQUENCE  
{  
target GenericObjectReference,  
open-succeeded ObjectReference,  
protocol GenericOctetString,  
address GenericOctetString,  
connection-tag GenericInteger  
}
```

```
PutBefore ::= SEQUENCE
```

```
{  
  target GenericObjectReference,  
  reference-visible GenericObjectReference  
}
```

PutBehind ::= SEQUENCE

```
{  
  target GenericObjectReference,  
  reference-visible GenericObjectReference  
}
```

ReadPersistent ::= SEQUENCE

```
{  
  target GenericObjectReference,  
  read-succeeded ObjectReference,  
  out-variables SEQUENCE SIZE (1..MAX) OF ObjectReference,  
  in-file-name GenericOctetString  
}
```

ScaleBitmap ::= SEQUENCE

```
{  
  target GenericObjectReference,  
  x-scale GenericInteger,  
  y-scale GenericInteger  
}
```

ScaleVideo ::= SEQUENCE

```
{  
  target GenericObjectReference,  
  x-scale GenericInteger,  
  y-scale GenericInteger  
}
```

ScrollItems ::= SEQUENCE

```
{  
  target GenericObjectReference,  
  items-to-scroll GenericInteger  
}
```

SelectItem ::= SEQUENCE

```
{  
  target GenericObjectReference,  
  item-index GenericInteger  
}
```

```
SendEvent ::= SEQUENCE
{
    target GenericObjectReference,
    emulated-event-source GenericObjectReference,
    emulated-event-type EventType,
    emulated-event-data EmulatedEventData OPTIONAL
}
```

```
SetBoxSize ::= SEQUENCE
{
    target GenericObjectReference,
    x-new-box-size GenericInteger,
    y-new-box-size GenericInteger
}
```

```
SetCachePriority ::= SEQUENCE
{
    target GenericObjectReference,
    new-cache-priority GenericInteger
}
```

```
SetCounterEndPosition ::= SEQUENCE
{
    target GenericObjectReference,
    new-counter-end-position GenericInteger
}
```

```
SetCounterPosition ::= SEQUENCE
{
    target GenericObjectReference,
    new-counter-position GenericInteger
}
```

```
SetCounterTrigger ::= SEQUENCE
{
    target GenericObjectReference,
    trigger-identifier GenericInteger,
    new-counter-value GenericInteger OPTIONAL
}
```

```
SetCursorPosition ::= SEQUENCE
{
    target GenericObjectReference,
```

```
x-cursor GenericInteger,  
y-cursor GenericInteger  
}
```

```
SetCursorShape ::= SEQUENCE  
{  
    target GenericObjectReference,  
    new-cursor-shape GenericObjectReference OPTIONAL  
}
```

```
SetData ::= SEQUENCE  
{  
    target GenericObjectReference,  
    new-content NewContent  
}
```

```
SetEntryPoint ::= SEQUENCE  
{  
    target GenericObjectReference,  
    new-entry-point GenericInteger  
}
```

```
SetFillColorur ::= SEQUENCE  
{  
    target GenericObjectReference,  
    new-fill-colour NewColour OPTIONAL  
}
```

```
SetFirstItem ::= SEQUENCE  
{  
    target GenericObjectReference,  
    new-first-item GenericInteger  
}
```

```
SetFontRef ::= SEQUENCE  
{  
    target GenericObjectReference,  
    new-font NewFont  
}
```

```
SetHighlightStatus ::= SEQUENCE  
{  
    target GenericObjectReference,  
    new-highlight-status GenericBoolean  
}
```

```
}
```

```
SetInteractionStatus ::= SEQUENCE
{
    target GenericObjectReference,
    new-interaction-status GenericBoolean
}
```

```
SetLabel ::= SEQUENCE
{
    target GenericObjectReference,
    new-label GenericOctetString
}
```

```
SetLineColour ::= SEQUENCE
{
    target GenericObjectReference,
    new-line-colour NewColour
}
```

```
SetLineStyle ::= SEQUENCE
{
    target GenericObjectReference,
    new-line-style GenericInteger
}
```

```
SetLineWidth ::= SEQUENCE
{
    target GenericObjectReference,
    new-line-width GenericInteger
}
```

```
SetOverwriteMode ::= SEQUENCE
{
    target GenericObjectReference,
    new-overwrite-mode GenericBoolean
}
```

```
SetPaletteRef ::= SEQUENCE
{
    target GenericObjectReference,
    new-palette-ref GenericObjectReference
}
```

```
SetPortion ::= SEQUENCE
{
    target GenericObjectReference,
    new-portion GenericInteger
}
```

```
SetPosition ::= SEQUENCE
{
    target GenericObjectReference,
    new-x-position GenericInteger,
    new-y-position GenericInteger
}
```

```
SetSliderValue ::= SEQUENCE
{
    target GenericObjectReference,
    new-slider-value GenericInteger
}
```

```
SetSpeed ::= SEQUENCE
{
    target GenericObjectReference,
    new-speed Rational
}
```

```
SetTimer ::= SEQUENCE
{
    target GenericObjectReference,
    timer-id GenericInteger,
    new-timer NewTimer OPTIONAL
}
```

```
NewTimer ::= SEQUENCE
{
    timer-value GenericInteger,
    absolute-time GenericBoolean OPTIONAL
}
```

```
SetTransparency ::= SEQUENCE
{
    target GenericObjectReference,
    new-transparency GenericInteger
}
```

SetVariable ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    new-variable-value NewVariableValue  
}
```

SetVolume ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    new-volume GenericInteger  
}
```

Step ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    nb-of-steps GenericInteger  
}
```

StorePersistent ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    store-succeeded ObjectReference,  
    in-variables SEQUENCE SIZE (1..MAX) OF ObjectReference,  
    out-file-name GenericOctetString  
}
```

Subtract ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    value GenericInteger  
}
```

TestVariable ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    operator GenericInteger,  
    comparison-value ComparisonValue  
}
```

ToggleItem ::= SEQUENCE

```
{  
    target GenericObjectReference,  
    item-index GenericInteger  
}
```

TransitionTo ::= SEQUENCE

```
{
  target GenericObjectReference,
  connection-tag-or-null ConnectionTagOrNull,
  transition-effect GenericInteger OPTIONAL
}
```

ConnectionTagOrNull ::= CHOICE

```
{
  connection-tag GenericInteger,
  null NULL
}
```

ComparisonValue ::= CHOICE

```
{
  new-generic-boolean [225] GenericBoolean,
  new-generic-integer [226] GenericInteger,
  new-generic-octetstring [227] GenericOctetString,
  new-generic-object-reference [228] GenericObjectReference,
  new-generic-content-reference [229] GenericContentReference
}
```

EmulatedEventData ::= CHOICE

```
{
  new-generic-boolean [225] GenericBoolean,
  new-generic-integer [226] GenericInteger,
  new-generic-octet-string [227] GenericOctetString
}
```

NewColour ::= CHOICE

```
{
  new-colour-index [230] GenericInteger,
  new-absolute-colour [231] GenericOctetString
}
```

NewContent ::= CHOICE

```
{
  new-included-content GenericOctetString,
  new-referenced-content NewReferencedContent
}
```

NewFont ::= CHOICE

```
{
```

208

```
new-font-name [232] GenericOctetString,  
new-font-reference [233] GenericObjectReference  
}
```

```
NewReferencedContent ::= SEQUENCE  
{  
    generic-content-reference GenericContentReference,  
    new-content-size [234] NewContentSize,  
    new-content-cache-priority [235] GenericInteger OPTIONAL  
}
```

```
NewContentSize ::= CHOICE  
{  
    content-size GenericInteger,  
    null NULL  
}
```

```
NewVariableValue ::= CHOICE  
{  
    new-generic-integer [226] GenericInteger,  
    new-generic-boolean [225] GenericBoolean,  
    new-generic-octet-string [227] GenericOctetString,  
    new-generic-object-reference [228] GenericObjectReference,  
    new-generic-content-reference [229] GenericContentReference  
}
```

```
Parameter ::= CHOICE  
{  
    new-generic-boolean [225] GenericBoolean,  
    new-generic-integer [226] GenericInteger,  
    new-generic-octetstring [227] GenericOctetString,  
    new-generic-object-reference [228] GenericObjectReference,  
    new-generic-content-reference [229] GenericContentReference  
}
```

```
Point ::= SEQUENCE  
{  
    x GenericInteger,  
    y GenericInteger  
}
```

```
Rational ::= SEQUENCE  
{  
    numerator GenericInteger,
```

```

    denominator GenericInteger OPTIONAL
}

```

-- A.43 Referencing Objects, Contents, Values, Colour and Position ____

```

ObjectReference ::= CHOICE
{
    external-reference ExternalReference,
    internal-reference INTEGER (1..MAX)
}

```

```

ExternalReference ::= SEQUENCE
{
    group-identifier OCTET STRING,
    object-number INTEGER (0..MAX)
}

```

```

IndirectReference ::= [236] ObjectReference

```

```

ContentReference ::= OCTET STRING

```

```

GenericObjectReference ::= CHOICE
{
    direct-reference ObjectReference,
    indirect-reference IndirectReference
}

```

```

GenericContentReference ::= CHOICE
{
    content-reference [69] ContentReference,
    indirect-reference IndirectReference
}

```

```

GenericInteger ::= CHOICE
{
    integer INTEGER,
    indirect-reference IndirectReference
}

```

```

GenericBoolean ::= CHOICE
{
    boolean BOOLEAN,
    indirect-reference IndirectReference
}

```

```
GenericOctetString ::= CHOICE
{
    octetstring OCTET STRING,
    indirect-reference IndirectReference
}
```

```
Colour ::= CHOICE
{
    colour-index INTEGER,
    absolute-colour OCTET STRING
}
```

```
XYPosition ::= SEQUENCE
{
    x-position INTEGER,
    y-position INTEGER
}
```

```
END
```

Annex B

(normative)

Textual notation for MHEG-5 applications

This annex describes the textual notation of MHEG-5 objects conforming to ISO/IEC 13522-5. The alternative representation is the ASN.1 notation defined in Annex A.

B.1 General Definitions

B.1.1 Code

The code set to be used by the textual notation shall be ISO/IEC.646:1991 "Information technology -- ISO 7-bit coded character set for information interchange". And the textual notation shall use a subset of ISO/IEC 646, which shall be the range of characters from 0x20 (SP) to 0x7c (~), plus 0x09 (HT), 0x0a (LF), 0x0c (FF), and 0x0d (CR).

Other characters shall not be used.

NOTES

- 1 Despite the fact that the textual notation limits the character codes to be used, contents of 8 bit data can be encoded by means of QPRINTABLE (see B.3.4), BASE64 (see B.3.5) and external contents referenced by ContentReference.
- 2 An application domain could extend the character codes to be used in the textual notation as far as it doesn't violate the grammar. For example, characters from 0x80 to 0xfe might also be allowed for STRING and comments.

B.1.2 Delimiter

0x09 (HT), 0x0a (LF), 0x0c (FF), 0x0d (CR) and 0x20 (SP) are called delimiters.

The grammar described by the textual notation is word-based. A word is either a parenthesis ("(" or ")"), a brace ("{" or "}"), a tag (see B.1.4) or a terminal symbol (see B.3). Any number of delimiters may be inserted between any two adjacent words, without changing the interpretation of words. However, at least one delimiter shall exist between any two terminal symbols and between any tag and any terminal symbol, since they would otherwise be interpreted as a single terminal or a single tag.

B.1.3 Comment

// (0x2f 0x2f) which is not within a STRING, QPRINTABLE and BASE64 (see B.3.3, B.3.4 and B.3.5) is used to indicate the start of a comment. All characters between such // (including the //) and the next occurrence of a 0x0a (LF), 0x0c (FF) or 0x0d (CR) shall be ignored.

NOTE - 0x09 (HT) and 0x20 (SP) don't indicate the end of comments.

B.1.4 Tag

A token starting with : (0x3a) is called "tag". A tag is preceded by { (0x7b) when it is used at the beginning of MHEG-5 objects. A tag is used to distinguish the MHEG-5 objects and their associated attribute values, in general. Tags are case insensitive, e.g. ":Root", ":root", ":ROOT", ":rOOt" and so on are all same. However, in this textual notation, some combination of upper case and lower case in tags are used for easier understanding and more readability.

B.2 Definitions of Symbols

The following Table B.1 shows the symbols used in the textual notation and their meanings.

Symbol	Definition
::=	Is defined to be.
	Alternative.
<">	Double quote mark (0x22).
"text"	Literals enclosed in double quotes.
<text>	Plain text description explaining the codes to be here.
*	The preceding syntactic unit may be repeated zero or more times.
+	The preceding syntactic unit may be repeated one or more times.
[]	The enclosed syntactic unit is optional. It may occur zero or one times.
.	End of clause.

Table B.1 Definitions of Symbols in Textual Notation

B.3 Terminal Symbols

All the terminal symbols used in the textual notation are defined as follows.

B.3.1 INTEGER

A decimal or positive hexadecimal integer value.

Definition:

```

INTEGER      ::= DECINT | HEXINT | "0" .
DECINT       ::= ["-"] DIGIT [DIGIT0]* .
DIGIT        ::= "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" .
DIGIT0       ::= DIGIT | "0" .
HEXINT       ::= HEXPREFIX HEXDIGIT0 [HEXDIGIT0]* .
HEXPREFIX    ::= "0x" | "0X" .
HEXDIGIT0    ::= DIGIT | "0" | "a" | "b" | "c" | "d" | "e" | "f" | "A" | "B" | "C" | "D" | "E" | "F" .

```

B.3.2 BOOLEAN

A Boolean value may be either true or false. The BOOLEAN terminal is case insensitive, i.e. "True", "TRUE" and "true" are one and the same, "False", "FALSE" and "false" are one and the same.

Definition:

```

BOOLEAN      ::= "true" | "false" .

```

B.3.3 STRING

A string value enclosed in the double quotes may contain an arbitrary number of printable characters (from 0x20 to 0x7e). A double quote (0x22) within a STRING shall be encoded as \" (0x5c 0x22). And a backslash (0x5c) shall be encoded as \\ (0x5c 0x5c).

Note that no line breaks are not to be included in STRING, i.e. multi line text content shall be encoded using QPRINTABLE or BASE64.

Definition:

STRING ::= <"> STRINGCHAR* <"> .
 STRINGCHAR ::= <any single printable character except <"> and "\"> | "\" | "\\\" .

B.3.4 QPRINTABLE

A string value enclosed in the single quotes shall contain a Quoted-Printable encoded content as defined in RFC-1521. However, ' (0x27) shall be encoded as =27. The lower case characters such as a, b, c, d, e and f may be used as a general 8-bit representation defined by section 5.1 rule #1 in RFC-1521. The number of characters in a line is not limited. Line breaks need not to be converted to CR (0x0d)/LF (0x0a), however, at least one of LF (0x0a), FF (0x0c) and CR (0x0d) shall be used representing a line break.

Definition:

QPRINTABLE ::= "" QPRINTABLECHAR* "" .
 QPRINTABLECHAR ::= <a character as defined above> | <encoding sequence of a character as defined above> .

B.3.5 BASE64

A string value enclosed in the back quotes shall contain a BASE64 encoded content as defined in RFC-1521. However, the number of characters in a line is not limited in this standard. The encoded BASE64 content may be split into several segments by at least one of LF (0x0a), FF (0x0c) or CR (0x0d). However, these characters shall be ignored, and the input BASE64 encoded segments shall be handled as if they were in one line.

Definition:

BASE64 ::= "" BASE64CHAR* "" .
 BASE64CHAR ::= <an encoding sequence of a character as defined above> .

B.3.6 Null

Null represents a special terminal whose semantics depends on the MHEG-5 objects definition. The Null terminal is case insensitive, i.e. "NULL" and "null" are one and the same.

Definition:

Null ::= "NULL"

B.3.7 Enumeration Values

A terminal word starting with an alphabet character is called "enumeration value" (all enumeration values are enclosed within two double quotations in the following grammar). An enumeration value is used as one of the terminal symbols which represents a specific semantics depending on its usage. All enumeration values are case insensitive, i.e. "IsAvailable" and "isavailable" are same.

B.4 MHEG-5 Object Definitions

The textual notation of MHEG 5 objects are defined as follows.

Table B.2 summarises the abbreviations used in tags and made them in reasonable lengths while keeping the readability.

Bordered Bounding Box	BBBox
Content Cache Priority	CCPriority
Content Hook	CHook
Coordinate System	CS
Generic	G
Group Cache Priority	GCPriority
Horizontal Justification	HJustification
Identifier	ID
Information	Info
Interchanged	Interchg
OctetString	OString
Original	Orig
Program	Prg
Reference	Ref
Register	Reg
Standard	Std
Variable	Var (except for elementary action tags)
Vertical Justification	VJustification

Table B.2: Tag abbreviations

B.4.1 Root Class

Root ::= ObjectIdentifier .
ObjectIdentifier ::= ObjectReference .

B.4.2 Group Class

Group ::= Root [StandardIdentifier]
 [StandardVersion] [ObjectInformation]
 [OnStartUp] [OnCloseDown]
 [OriginalGroupCachePriority] [Items] .
StandardIdentifier ::= ":StdID" JointIsoltIdentifier
 MHEGStandardIdentifier .
JointIsoltIdentifier ::= INTEGER .
MHEGStandardIdentifier ::= INTEGER .
StandardVersion ::= ":StdVersion" INTEGER .
ObjectInformation ::= ":ObjectInfo" OctetString .
OnStartUp ::= ":OnStartUp" ActionClass .

```

OnCloseDown      ::= ":OnCloseDown" ActionClass .
OriginalGroupCachePriority ::= ":OrigGCPriority" INTEGER .
Items            ::= ":Items" "(" GroupItem+ ")" .
GroupItem        ::= ResidentProgramClass |
                    RemoteProgramClass |
                    InterchangedProgramClass |
                    PaletteClass |
                    FontClass |
                    CursorShapeClass |
                    BooleanVariableClass |
                    IntegerVariableClass |
                    OctetStringVariableClass |
                    ObjectRefVariableClass |
                    ContentRefVariableClass |
                    LinkClass |
                    StreamClass |
                    BitmapClass |
                    LineArtClass |
                    DynamicLineArtClass |
                    RectangleClass |
                    HotspotClass |
                    SwitchButtonClass |
                    PushButtonClass |
                    TextClass |
                    EntryFieldClass |
                    HyperTextClass |
                    SliderClass |
                    TokenGroupClass |
                    ListGroupClass .

```

B.4.3 Application Class

```

ApplicationClass ::= "{:Application" Group
                    [OnSpawnCloseDown] [OnRestart]
                    [DefaultAttributes] }" .
OnSpawnCloseDown ::= ":OnSpawnCloseDown" ActionClass .
OnRestart        ::= ":OnRestart" ActionClass .
DefaultAttributes ::= DefaultAttribute+ .
DefaultAttribute ::= CharacterSet | BackgroundColour
                    | TextContentHook
                    | TextColour | Font
                    | FontAttributes
                    | InterchangedProgramContentHook
                    | StreamContentHook

```

```

        | BitmapContentHook
        | LineArtContentHook | ButtonRefColour
        | HighlightRefColour | SliderRefColour .
CharacterSet ::= ":CharacterSet" INTEGER .
BackgroundColour ::= ":BackgroundColour" Colour .
TextContentHook ::= ":TextCHook" INTEGER .
TextColour ::= ":TextColour" Colour .
Font ::= ":Font" FontBody .
FontBody ::= DirectFont | IndirectFont .
DirectFont ::= OctetString .
IndirectFont ::= ObjectReference .
FontAttributes ::= ":FontAttributes" OctetString .
InterchangedProgramContentHook ::= ":InterchgPrgCHook" INTEGER .
StreamContentHook ::= ":StreamCHook" INTEGER .
BitmapContentHook ::= ":BitmapCHook" INTEGER .
LineArtContentHook ::= ":LineArtCHook" INTEGER .
ButtonRefColour ::= ":ButtonRefColour" Colour .
HighlightRefColour ::= ":HighlightRefColour" Colour .
SliderRefColour ::= ":SliderRefColour" Colour .

```

B.4.4 Scene Class

```

SceneClass ::= "{:Scene" Group InputEventRegister
            SceneCoordinateSystem [AspectRatio]
            [MovingCursor] [NextScenes] }" .
InputEventRegister ::= ":InputEventReg" INTEGER .
SceneCoordinateSystem ::= ":SceneCS" XScene YScene .
XScene ::= INTEGER .
YScene ::= INTEGER .
AspectRatio ::= ":AspectRatio" Width Height .
Width ::= INTEGER .
Height ::= INTEGER .
MovingCursor ::= ":MovingCursor" BOOLEAN .
NextScenes ::= ":NextScenes" "(" NextScene+ ")" .
NextScene ::= "(" SceneRef SceneWeight ")" .
SceneRef ::= OctetString .
SceneWeight ::= INTEGER .

```

B.4.5 Ingredient Class

```

Ingredient ::= Root [InitiallyActive] [ContentHook]
            [OriginalContent] [Shared] .
InitiallyActive ::= ":InitiallyActive" BOOLEAN .
ContentHook ::= ":CHook" INTEGER .
OriginalContent ::= ":OrigContent" ContentBody .

```

```

ContentBody      ::= IncludedContent | ReferencedContent .
IncludedContent  ::= OctetString .
ReferencedContent ::= ":"ContentRef" "(" ContentReference
                        [ContentSize] [ContentCachePriority]
                        ")" .
ContentSize      ::= ":"ContentSize" INTEGER .
ContentCachePriority ::= ":"CCPriority" INTEGER .
Shared           ::= ":"Shared" BOOLEAN .

```

B.4.6 Link Class

```

LinkClass ::= "{Link" Ingredient LinkCondition
           LinkEffect "}" .

LinkCondition ::= EventSource EventType [EventData] .

EventSource ::= ":"EventSource" ObjectReference .

EventType ::= ":"EventType" EventTypeEnum .

EventTypeEnum ::= "IsAvailable" | "ContentAvailable"
                | "IsDeleted" | "IsRunning"
                | "IsStopped" | "UserInput"
                | "AnchorFired" | "TimerFired"
                | "AsynchStopped" | "InteractionCompleted"
                | "TokenMovedFrom" | "TokenMovedTo"
                | "StreamEvent" | "StreamPlaying"
                | "StreamStopped" | "CounterTrigger"
                | "HighlightOn" | "HighlightOff"
                | "CursorEnter" | "CursorLeave"
                | "IsSelected" | "IsDeselected"
                | "TestEvent" | "FirstItemPresented"
                | "LastItemPresented" | "HeadItems"
                | "TailItems" | "ItemSelected"
                | "ItemDeselected" | "EntryFieldFull"
                | "EngineEvent" .

EventData ::= ":"EventData" EventDataBody .

EventDataBody ::= OctetString | BOOLEAN | INTEGER .

LinkEffect ::= ":"LinkEffect" ActionClass .

```

B.4.7 Program Class

```

Program          ::= Ingredient Name [InitiallyAvailable] .
Name             ::= ":Name" OctetString .
InitiallyAvailable ::= ":InitiallyAvailable" BOOLEAN .

```

B.4.8 ResidentProgram Class

```
ResidentProgramClass ::= "{:ResidentPrg" Program "}" .
```

B.4.9 RemoteProgram Class

```
RemoteProgramClass ::= "{:RemotePrg" Program
                    [ProgramConnectionTag] }" .
ProgramConnectionTag ::= ":ConnectionTag" INTEGER .
```

B.4.10 InterchangedProgram Class

```
InterchangedProgramClass ::= "{:InterchgPrg" Program "}" .
```

B.4.11 Palette Class

```
PaletteClass ::= "{:Palette" Ingredient "}" .
```

B.4.12 Font Class

```
FontClass ::= "{:Font" Ingredient "}" .
```

B.4.13 CursorShape Class

```
CursorShapeClass ::= "{:CursorShape" Ingredient "}" .
```

B.4.14 Variable Class

```

Variable                ::= Ingredient OriginalValue .
OriginalValue            ::= ":"OrigValue" OriginalValueBody .
OriginalValueBody        ::= BOOLEAN | INTEGER | OctetString
                             | ObjectReferenceValue
                             | ContentReferenceValue .
ObjectReferenceValue     ::= ":"ObjectRef" ObjectReference .
ContentReferenceValue    ::= ":"ContentRef" ContentReference .

```

B.4.15 BooleanVariable Class

```
BooleanVariableClass ::= "{:BooleanVar" Variable "}" .
```

B.4.16 IntegerVariable Class

```
IntegerVariableClass ::= "{:IntegerVar" Variable "}" .
```

B.4.17 OctetStringVariable Class

```
OctetStringVariableClass ::= "{:OStringVar" Variable "}" .
```

B.4.18 ObjectRefVariable Class

```
ObjectRefVariableClass ::= "{:ObjectRefVar" Variable "}" .
```

B.4.19 ContentRefVariable Class

```
ContentRefVariableClass ::= "{:ContentRefVar" Variable "}" .
```

B.4.20 Presentable Class

Presentable ::= Ingredient .

B.4.21 TokenManager Class

```

TokenManager      ::= [MovementTable] .
MovementTable     ::= ":"MovementTable" "(" Movement+ ")" .
Movement          ::= "(" TargetElement+ ")" .
TargetElement     ::= INTEGER .

```

B.4.22 TokenGroup Class

```

TokenGroupClass      ::= "{:TokenGroup" TokenGroupBody "}" .
TokenGroupBody      ::= Presentable TokenManager TokenGroupItems
                        [NoTokenActionSlots] .
TokenGroupItems      ::= ":{TokenGroupItems" "(" TokenGroupItem+ ")" .
TokenGroupItem       ::= "(" AVisible [ActionSlots] ")" .
AVisible             ::= ObjectReference .
ActionSlots          ::= ":{ActionSlots" "(" ActionSlot+ ")" .
ActionSlot           ::= ActionClass | Null .
NoTokenActionSlots   ::= ":{NoTokenActionSlots" "(" ActionSlot+ ")" .

```

B.4.23 ListGroup Class

```
ListGroupClass      ::= "{:ListGroup" TokenGroupBody
                        Positions [WrapAround]
                        [MultipleSelection] "}" .
Positions           ::= ":Positions" "(" Position+ ")" .
Position            ::= "(" XYPosition ")" .
WrapAround          ::= ":WrapAround" BOOLEAN .
MultipleSelection   ::= ":MultipleSelection" BOOLEAN .
```

B.4.24 Visible Class

```
Visible          ::= Presentable OriginalBoxSize
                  [OriginalPosition] [OriginalPaletteRef] .
OriginalBoxSize  ::= "OrigBoxSize" BoxSize .
BoxSize         ::= XLength YLength .
XLength         ::= INTEGER .
YLength         ::= INTEGER .
OriginalPosition ::= ":OrigPosition" XYPosition .
OriginalPaletteRef ::= ":OrigPaletteRef" ObjectReference .
```

B.4.25 Bitmap Class

```
BitmapClass ::= "{:Bitmap" Visible [Tiling]
               [OriginalTransparency] "}" .
```

Tiling ::= ":Tiling" BOOLEAN .
 OriginalTransparency ::= ":OrigTransparency" INTEGER .

B.4.26 LineArt Class

LineArtClass ::= "{:LineArt" LineArtBody "}" .
 LineArtBody ::= Visible [BorderedBoundingBox]
 [OriginalLineWidth]
 [OriginalLineStyle]
 [OriginalRefLineColour]
 [OriginalRefFillColor] "}" .
 BorderedBoundingBox ::= ":BBBox" BOOLEAN .
 OriginalLineWidth ::= ":OrigLineWidth" INTEGER .
 OriginalLineStyle ::= ":OrigLineStyle" INTEGER .
 OriginalRefLineColour ::= ":OrigRefLineColour" Colour .
 OriginalRefFillColor ::= ":OrigRefFillColor" Colour .

B.4.27 Rectangle Class

RectangleClass ::= "{:Rectangle" LineArtBody "}" .

B.4.28 DynamicLineArt Class

DynamicLineArtClass ::= "{:DynamicLineArt" LineArtBody "}" .

B.4.29 Text Class

TextClass ::= "{:Text" TextBody "}" .
 TextBody ::= Visible [OriginalFont] [FontAttributes]
 [TextColour] [BackgroundColour]
 [CharacterSet]
 [HorizontalJustification]
 [VerticalJustification]
 [LineOrientation] [StartCorner]
 [TextWrapping] .
 OriginalFont ::= ":OrigFont" FontBody .
 HorizontalJustification ::= ":HJustification" JustificationEnum .
 JustificationEnum ::= "start" | "end" | "centre" | "justified" .
 VerticalJustification ::= ":VJustification" JustificationEnum .
 LineOrientation ::= ":LineOrientation" LineOrientationEnum .
 LineOrientationEnum ::= "vertical" | "horizontal" .
 StartCorner ::= ":StartCorner" StartCornerEnum .
 StartCornerEnum ::= "upper-left" | "upper-right"
 | "lower-left" | "lower-right" .
 TextWrapping ::= ":TextWrapping" BOOLEAN .

B.4.30 Stream Class

```

StreamClass      ::= "{:Stream" Presentable Multiplex
                    [Storage] [Looping] "}" .
Multiplex        ::= ":Multiplex" "(" StreamComponent+ ")" .
StreamComponent  ::= AudioClass | VideoClass | RTGraphicsClass .
Storage          ::= ":Storage" StorageEnum .
StorageEnum      ::= "memory" | "stream" .
Looping          ::= ":Looping" INTEGER .

```

B.4.31 Audio Class

```

AudioClass       ::= "{:Audio" Presentable ComponentTag
                    [OriginalVolume] "}" .
ComponentTag     ::= ":ComponentTag" INTEGER .
OriginalVolume   ::= ":OrigVolume" INTEGER .

```

B.4.32 Video Class

```

VideoClass       ::= "{:Video" Visible ComponentTag
                    [Termination] "}" .
Termination      ::= ":Termination" TerminationEnum .
TerminationEnum  ::= "freeze" | "disappear" .

```

B.4.33 RTGraphics Class

```

RTGraphicsClass  ::= "{:RTGraphics" Visible ComponentTag
                    [Termination] "}" .

```

B.4.34 Interactable Class

```

Interactable     ::= [EngineResp] [HighlightRefColour] .
EngineResp       ::= ":EngineResp" BOOLEAN .

```

B.4.35 Slider Class

```

SliderClass      ::= "{:Slider" Visible Interactable
                    Orientation MaxValue [MinValue]
                    [InitialValue] [InitialPortion]
                    [StepSize] [SliderStyle]
                    [SliderRefColour] "}" .
Orientation       ::= ":Orientation" OrientationEnum .
OrientationEnum   ::= "left" | "right" | "up" | "down" .
MaxValue          ::= ":MaxValue" INTEGER .
MinValue          ::= ":MinValue" INTEGER .
InitialValue      ::= ":InitialValue" INTEGER .
InitialPortion    ::= ":InitialPortion" INTEGER .
StepSize         ::= ":StepSize" INTEGER .

```

SliderStyle ::= ":SliderStyle" SliderStyleEnum .
 SliderStyleEnum ::= "normal" | "thermometer" | "proportional" .

B.4.36 EntryField Class

EntryFieldClass ::= "{:EntryField" TextBody Interactable
 [InputType] [CharList]
 [ObscuredInput] [MaxLength] }" .
 InputType ::= ":InputType" InputTypeEnum .
 InputTypeEnum ::= "alpha" | "numeric" | "any" | "listed" .
 CharList ::= ":CharList" OctetString .
 ObscuredInput ::= ":ObscuredInput" BOOLEAN .
 MaxLength ::= ":MaxLength" INTEGER .

B.4.37 HyperText Class

HyperTextClass ::= "{:HyperText" TextBody Interactable
 }" .

B.4.38 Button Class

Button ::= Visible Interactable [ButtonRefColour] .

B.4.39 Hotspot Class

HotspotClass ::= "{:Hotspot" Button }" .

B.4.40 PushButton Class

PushButtonClass ::= "{:PushButton" PushButtonBody }" .
 PushButtonBody ::= Button [OriginalLabel] [CharacterSet] .
 OriginalLabel ::= ":OrigLabel" OctetString .

B.4.41 SwitchButton Class

SwitchButtonClass ::= "{:SwitchButton" PushButtonBody
 ButtonStyle }" .
 ButtonStyle ::= ":ButtonStyle" ButtonStyleEnum .
 ButtonStyleEnum ::= "pushbutton" | "radiobutton" | "checkbox" .

B.4.42 Action Class

ActionClass ::= "(" ElementaryAction+ ")" .
 ElementaryAction ::= Activate
 | Add
 | AddItem
 | Append
 | BringToFront
 | Call

- | CallActionSlot
- | Clear
- | Clone
- | CloseConnection
- | Deactivate
- | DelItem
- | Deselect
- | DeselectItem
- | Divide
- | DrawArc
- | DrawLine
- | DrawOval
- | DrawPolygon
- | DrawPolyline
- | DrawRectangle
- | DrawSector
- | Fork
- | GetAvailabilityStatus
- | GetBoxSize
- | GetCellItem
- | GetCursorPosition
- | GetEngineSupport
- | GetEntryPoint
- | GetFillColor
- | GetFirstItem
- | GetHighlightStatus
- | GetInteractionStatus
- | GetItemStatus
- | GetLabel
- | GetLastAnchorFired
- | GetLineColour
- | GetLineStyle
- | GetLineWidth
- | GetListItem
- | GetListSize
- | GetOverwriteMode
- | GetPortion
- | GetPosition
- | GetRunningStatus
- | GetSelectionStatus
- | GetSliderValue
- | GetTextContent
- | GetTextData
- | GetTokenPosition

- | GetVolume
- | Launch
- | LockScreen
- | Modulo
- | Move
- | MoveTo
- | Multiply
- | OpenConnection
- | Preload
- | PutBefore
- | PutBehind
- | Quit
- | ReadPersistent
- | Run
- | ScaleBitmap
- | ScaleVideo
- | ScrollItems
- | Select
- | SelectItem
- | SendEvent
- | SendToBack
- | SetBoxSize
- | SetCachePriority
- | SetCounterEndPosition
- | SetCounterPosition
- | SetCounterTrigger
- | SetCursorPosition
- | SetCursorShape
- | SetData
- | SetEntryPoint
- | SetFillColor
- | SetFirstItem
- | SetFontRef
- | SetHighlightStatus
- | SetInteractionStatus
- | SetLabel
- | SetLineColour
- | SetLineStyle
- | SetLineWidth
- | SetOverwriteMode
- | SetPaletteRef
- | SetPortion
- | SetPosition
- | SetSliderValue

| SetSpeed
 | SetTimer
 | SetTransparency
 | SetVariable
 | SetVolume
 | Spawn
 | Step
 | Stop
 | StorePersistent
 | Subtract
 | TestVariable
 | Toggle
 | ToggleItem
 | TransitionTo
 | Unload
 | UnlockScreen .

Activate ::= ":Activate" "(" Target ")" .
 Add ::= ":Add" "(" Target Value ")" .
 AddItem ::= ":AddItem" "(" Target ItemIndex
 VisibleReference ")" .
 Append ::= ":Append" "(" Target AppendValue ")" .
 BringToFront ::= ":BringToFront" "(" Target ")" .
 Call ::= ":Call" "(" Target CallSucceeded
 [Parameters] ")" .
 CallActionSlot ::= ":CallActionSlot" "(" Target Index ")" .
 Clear ::= ":Clear" "(" Target ")" .
 Clone ::= ":Clone" "(" Target CloneRefVar ")" .
 CloseConnection ::= ":CloseConnection" "(" Target
 ConnectionTag ")" .
 Deactivate ::= ":Deactivate" "(" Target ")" .
 DelItem ::= ":DelItem" "(" Target VisibleReference
 ")" .
 Deselect ::= ":Deselect" "(" Target ")" .
 DeselectItem ::= ":DeselectItem" "(" Target ItemIndex ")" .
 Divide ::= ":Divide" "(" Target Value ")" .
 DrawArc ::= ":DrawArc" "(" Target X Y EllipseWidth
 EllipseHeight StartAngle ArcAngle ")" .
 DrawLine ::= ":DrawLine" "(" Target X1 Y1 X2 Y2 ")" .
 DrawOval ::= ":DrawOval" "(" Target X Y EllipseWidth
 EllipseHeight ")" .
 DrawPolygon ::= ":DrawPolygon" "(" Target PointList ")" .
 DrawPolyline ::= ":DrawPolyline" "(" Target PointList ")" .
 DrawRectangle ::= ":DrawRectangle" "(" Target X1 Y1 X2 Y2

```

        ")" .
DrawSector      ::= ":DrawSector" "(" Target X Y EllipseWidth
                    EllipseHeight StartAngle ArcAngle ")" .
Fork            ::= ":Fork" "(" Target ForkSucceeded
                    [Parameters] ")" .
GetAvailabilityStatus ::= ":GetAvailabilityStatus" "(" Target
                    AvailabilityStatusVar ")" .
GetBoxSize      ::= ":GetBoxSize" "(" Target XBoxSizeVar
                    YBoxSizeVar ")" .
GetCellItem     ::= ":GetCellItem" "(" Target CellIndex
                    ItemRefVar ")" .
GetCursorPosition ::= ":GetCursorPosition" "(" Target XOut YOut
                    ")" .
GetEngineSupport ::= ":GetEngineSupport" "(" Target Feature
                    Answer ")" .
GetEntryPoint   ::= ":GetEntryPoint" "(" Target EntryPointVar
                    ")" .
GetFillColour   ::= ":GetFillColour" "(" Target FillColourVar
                    ")" .
GetFirstItem    ::= ":GetFirstItem" "(" Target FirstItemVar
                    ")" .
GetHighlightStatus ::= ":GetHighlightStatus" "(" Target
                    HighlightStatusVar ")" .
GetInteractionStatus ::= ":GetInteractionStatus" "(" Target
                    InteractionStatusVar ")" .
GetItemStatus   ::= ":GetItemStatus" "(" Target
                    ItemIndex ItemStatusVar ")" .
GetLabel        ::= ":GetLabel" "(" Target LabelVar ")" .
GetLastAnchorFired ::= ":GetLastAnchorFired" "(" Target
                    LastAnchorFiredVar ")" .
GetLineColour   ::= ":GetLineColour" "(" Target LineColourVar
                    ")" .
GetLineStyle    ::= ":GetLineStyle" "(" Target LineStyleVar
                    ")" .
GetLineWidth    ::= ":GetLineWidth" "(" Target LineWidthVar
                    ")" .
GetListItem     ::= ":GetListItem" "(" Target ItemIndex
                    ItemRefVar ")" .
GetListSize     ::= ":GetListSize" "(" Target SizeVar ")" .
GetOverwriteMode ::= ":GetOverwriteMode" "(" Target
                    OverwriteModeVar ")" .
GetPortion      ::= ":GetPortion" "(" Target PortionVar ")" .
GetPosition     ::= ":GetPosition" "(" Target XPositionVar
                    YPositionVar ")" .

```

```

GetRunningStatus ::= ":GetRunningStatus" "(" Target
                  RunningStatusVar ")" .
GetSelectionStatus ::= ":GetSelectionStatus" "(" Target
                      SelectionStatusVar ")" .
GetSliderValue ::= ":GetSliderValue" "(" Target
                  SliderValueVar ")" .
GetTextContent ::= ":GetTextContent" "(" Target
                  TextContentVar ")" .
GetTextData ::= ":GetTextData" "(" Target TextDataVar ")" .
GetTokenPosition ::= ":GetTokenPosition" "(" Target
                    TokenPositionVar ")" .
GetVolume ::= ":GetVolume" "(" Target VolumeVar ")" .
Launch ::= ":Launch" "(" Target ")" .
LockScreen ::= ":LockScreen" "(" Target ")" .
Modulo ::= ":Modulo" "(" Target Value ")" .
Move ::= ":Move" "(" Target MovementIdentifier ")" .
MoveTo ::= ":MoveTo" "(" Target Index ")" .
Multiply ::= ":Multiply" "(" Target Value ")" .
OpenConnection ::= ":OpenConnection" "(" Target OpenSucceeded
                  Protocol Address ConnectionTag ")" .
Preload ::= ":Preload" "(" Target ")" .
PutBefore ::= ":PutBefore" "(" Target ReferenceVisible
              ")" .
PutBehind ::= ":PutBehind" "(" Target ReferenceVisible
              ")" .
Quit ::= ":Quit" "(" Target ")" .
ReadPersistent ::= ":ReadPersistent" "(" Target ReadSucceeded
                  OutVariables InFileName ")" .
Run ::= ":Run" "(" Target ")" .
ScaleBitmap ::= ":ScaleBitmap" "(" Target XScale YScale
              ")" .
ScaleVideo ::= ":ScaleVideo" "(" Target XScale YScale ")" .
ScrollItems ::= ":ScrollItems" "(" Target ItemsToScroll ")" .
Select ::= ":Select" "(" Target ")" .
SelectItem ::= ":SelectItem" "(" Target ItemIndex ")" .
SendEvent ::= ":SendEvent" "(" Target EmulatedEventSource
                  EmulatedEventType [EmulatedEventData]
              ")" .
SendToBack ::= ":SendToBack" "(" Target ")" .
SetBoxSize ::= ":SetBoxSize" "(" Target XNewBoxSize
                  YNewBoxSize ")" .
SetCachePriority ::= ":SetCachePriority" "(" Target
                  NewCachePriority ")" .
SetCounterEndPosition ::= ":SetCounterEndPosition" "(" Target

```

```

NewCounterEndPosition ")".
SetCounterPosition ::= ":SetCounterPosition" "(" Target
NewCounterPosition ")".
SetCounterTrigger ::= ":SetCounterTrigger" "(" Target
TriggerIdentifier [NewCounterValue]
")".
SetCursorPosition ::= ":SetCursorPosition" "(" Target XCursor
YCursor ")".
SetCursorShape ::= ":SetCursorShape" "(" Target
[NewCursorShape ")".
SetData ::= ":SetData" "(" Target NewContent ")".
SetEntryPoint ::= ":SetEntryPoint" "(" Target NewEntryPoint
")".
SetFillColor ::= ":SetFillColor" "(" Target [NewColour]
")".
SetFirstItem ::= ":SetFirstItem" "(" Target NewFirstItem
")".
SetFontRef ::= ":SetFontRef" "(" Target NewFont ")".
SetHighlightStatus ::= ":SetHighlightStatus" "(" Target
NewHighlightStatus ")".
SetInteractionStatus ::= ":SetInteractionStatus" "(" Target
NewInteractionStatus ")".
SetLabel ::= ":SetLabel" "(" Target NewLabel ")".
SetLineColour ::= ":SetLineColour" "(" Target NewColour ")".
SetLineStyle ::= ":SetLineStyle" "(" Target NewLineStyle
")".
SetLineWidth ::= ":SetLineWidth" "(" Target NewLineWidth
")".
SetOverwriteMode ::= ":SetOverwriteMode" "(" Target
NewOverwriteMode ")".
SetPaletteRef ::= ":SetPaletteRef" "(" Target NewPaletteRef
")".
SetPortion ::= ":SetPortion" "(" Target NewPortion ")".
SetPosition ::= ":SetPosition" "(" Target NewXPosition
NewYPosition ")".
SetSliderValue ::= ":SetSliderValue" "(" Target
NewSliderValue ")".
SetSpeed ::= ":SetSpeed" "(" Target NewSpeed ")".
SetTimer ::= ":SetTimer" "(" Target TimerID
[TimerValue] [AbsoluteTime] ")".
SetTransparency ::= ":SetTransparency" "(" Target
NewTransparency ")".
SetVariable ::= ":SetVariable" "(" Target
NewVariableValue ")".

```

SetVolume ::= ":SetVolume" "(" Target NewVolume ")" .
 Spawn ::= ":Spawn" "(" Target ")" .
 Stop ::= ":Stop" "(" Target ")" .
 Step ::= ":Step" "(" Target NbOfSteps ")" .
 StorePersistent ::= ":StorePersistent" "(" Target
 StoreSucceeded InVariables OutFileName
 ")" .
 Subtract ::= ":Subtract" "(" Target Value ")" .
 TestVariable ::= ":TestVariable" "(" Target Operator
 ComparisonValue ")" .
 Toggle ::= ":Toggle" "(" Target ")" .
 ToggleItem ::= ":ToggleItem" "(" Target ItemIndex ")" .
 TransitionTo ::= ":TransitionTo" "(" Target [ConnectionTag]
 [TransitionEffect] ")" .
 Unload ::= ":Unload" "(" Target ")" .
 UnlockScreen ::= ":UnlockScreen" "(" Target ")" .

 AbsoluteTime ::= ":AbsoluteTime" GenericBoolean .
 Address ::= GenericOctetString .
 Answer ::= ObjectReference .
 AppendValue ::= GenericOctetString .
 ArcAngle ::= GenericInteger .
 AvailabilityStatusVar ::= ObjectReference .
 CallSucceeded ::= ObjectReference .
 CellIndex ::= GenericInteger .
 CloneRefVar ::= ObjectReference .
 ComparisonValue ::= NewGenericBoolean | NewGenericInteger
 | NewGenericOctetString
 | NewGenericObjectReference
 | NewGenericContentReference .
 ConnectionTag ::= ":ConnectionTag" GenericInteger .
 Denominator ::= GenericInteger .
 EllipseHeight ::= GenericInteger .
 EllipseWidth ::= GenericInteger .
 EmulatedEventData ::= NewGenericBoolean | NewGenericInteger
 | NewGenericOctetString .
 EmulatedEventSource ::= GenericObjectReference .
 EmulatedEventType ::= EventTypeEnum .
 EntryPointVar ::= ObjectReference .
 ForkSucceeded ::= ObjectReference .
 Feature ::= GenericOctetString .
 FillColourVar ::= ObjectReference .
 FirstItemVar ::= ObjectReference .
 HighlightStatusVar ::= ObjectReference .

Index ::= GenericInteger .
 InFileName ::= GenericOctetString .
 InteractionStatusVar ::= ObjectReference .
 InVariables ::= "(" ObjectReference+ ")" .
 ItemIndex ::= GenericInteger .
 ItemRefVar ::= ObjectReference .
 ItemStatusVar ::= ObjectReference .
 ItemsToScroll ::= GenericInteger .
 LabelVar ::= ObjectReference .
 LastAnchorFiredVar ::= ObjectReference .
 LineColourVar ::= ObjectReference .
 LineStyleVar ::= ObjectReference .
 LineWidthVar ::= ObjectReference .
 MovementIdentifier ::= GenericInteger .
 NbOfSteps ::= GenericInteger .
 NewAbsoluteColour ::= ":NewAbsoluteColour" GenericOctetString .
 NewCachePriority ::= GenericInteger .
 NewColour ::= NewColourIndex | NewAbsoluteColour .
 NewColourIndex ::= ":NewColourIndex" GenericInteger .
 NewContent ::= NewIncludedContent | NewReferencedContent .
 NewContentCachePriority ::= ":NewCCPriority" GenericInteger .
 NewCounterEndPosition ::= GenericInteger .
 NewCounterPosition ::= GenericInteger .
 NewContentSize ::= ":NewContentSize" GenericInteger .
 NewCounterValue ::= GenericInteger .
 NewCursorShape ::= GenericObjectReference .
 NewEntryPoint ::= GenericInteger .
 NewFirstItem ::= GenericInteger .
 NewFont ::= NewFontName | NewFontReference .
 NewFontName ::= NewGenericOctetString .
 NewFontReference ::= NewGenericObjectReference .
 NewGenericBoolean ::= ":GBoolean" GenericBoolean .
 NewGenericInteger ::= ":GInteger" GenericInteger .
 NewGenericOctetString ::= ":GOctetString" GenericOctetString .
 NewGenericObjectReference ::= ":GObjectRef" GenericObjectReference .
 NewGenericContentReference ::= ":GContentRef" GenericContentReference .
 NewHighlightStatus ::= GenericBoolean .
 NewIncludedContent ::= GenericOctetString .
 NewInteractionStatus ::= GenericBoolean .
 NewLabel ::= GenericOctetString .
 NewLineStyle ::= GenericInteger .
 NewLineWidth ::= GenericInteger .
 NewOverwriteMode ::= GenericBoolean .
 NewPaletteRef ::= GenericObjectReference .

NewPortion ::= GenericInteger .
 NewReferencedContent ::= ":"NewRefContent" "(" GenericContentReference
 [NewContentSize]
 [NewContentCachePriority] ")" .
 NewSliderValue ::= GenericInteger .
 NewSpeed ::= Rational .
 NewTransparency ::= GenericInteger .
 NewVariableValue ::= NewGenericInteger | NewGenericBoolean
 | NewGenericOctetString
 | NewGenericObjectReference
 | NewGenericContentReference .
 NewVolume ::= GenericInteger .
 NewXPosition ::= GenericInteger .
 NewYPosition ::= GenericInteger .
 Numerator ::= GenericInteger .
 OpenSucceeded ::= ObjectReference .
 Operator ::= GenericInteger .
 OutFileName ::= GenericOctetString .
 OutVariables ::= "(" ObjectReference+ ")" .
 OverwriteModeVar ::= ObjectReference .
 Parameter ::= NewGenericBoolean | NewGenericInteger
 | NewGenericOctetString
 | NewGenericObjectReference
 | NewGenericContentReference .
 Parameters ::= Parameter+ .
 Point ::= "(" X Y ")" .
 PointList ::= "(" Point+ ")" .
 PortionVar ::= ObjectReference .
 Protocol ::= GenericOctetString .
 Rational ::= Numerator [Denominator] .
 ReadSucceeded ::= ObjectReference .
 ReferenceVisible ::= GenericObjectReference .
 RunningStatusVar ::= ObjectReference .
 SelectionStatusVar ::= ObjectReference .
 SizeVar ::= ObjectReference .
 SliderValueVar ::= ObjectReference .
 StartAngle ::= GenericInteger .
 StoreSucceeded ::= ObjectReference .
 Target ::= GenericObjectReference .
 TextContentVar ::= ObjectReference .
 TextDataVar ::= ObjectReference .
 TimerID ::= GenericInteger .
 TimerValue ::= GenericInteger .
 TokenPositionVar ::= ObjectReference .

TransitionEffect ::= GenericInteger .
 TriggerIdentifier ::= GenericInteger .
 Value ::= GenericInteger .
 VisibleReference ::= GenericObjectReference .
 VolumeVar ::= ObjectReference .
 X ::= GenericInteger .
 X1 ::= GenericInteger .
 X2 ::= GenericInteger .
 XBoxSizeVar ::= ObjectReference .
 XCursor ::= GenericInteger .
 XNewBoxSize ::= GenericInteger .
 XOut ::= ObjectReference .
 XPositionVar ::= ObjectReference .
 XScale ::= GenericInteger .
 Y ::= GenericInteger .
 Y1 ::= GenericInteger .
 Y2 ::= GenericInteger .
 YBoxSizeVar ::= ObjectReference .
 YCursor ::= GenericInteger .
 YNewBoxSize ::= GenericInteger .
 YOut ::= ObjectReference .
 YPositionVar ::= ObjectReference .
 YScale ::= GenericInteger .

B.4.43 Referencing Objects, Contents, Values, Colour and Position

ObjectReference ::= ExternalReference | InternalReference .
 ExternalReference ::= "(" GroupIdentifier ObjectNumber ")" .
 InternalReference ::= ObjectNumber .
 GroupIdentifier ::= OctetString .
 ObjectNumber ::= INTEGER .

 ContentReference ::= OctetString .

 GenericObjectReference ::= DirectReference | IndirectReference .
 DirectReference ::= ObjectReference .
 IndirectReference ::= ":" IndirectRef " ObjectReference .

 GenericContentReference ::= ContentReference | IndirectReference .

 GenericInteger ::= INTEGER | IndirectReference .

 GenericBoolean ::= BOOLEAN | IndirectReference .

GenericOctetString ::= OctetString | IndirectReference .

OctetString ::= STRING | QPRINTABLE | BASE64 .

Colour ::= ColourIndex | AbsoluteColour .

ColourIndex ::= INTEGER .

AbsoluteColour ::= OctetString .

XYPosition ::= XPosition YPosition .

XPosition ::= INTEGER .

YPosition ::= INTEGER .

Annex C

(informative)

Bootstrap of an MHEG-5 engine

An MHEG-5 engine may be in two main states: idle or active. An idle MHEG-5 engine is one in which no Application object is active. Conversely, in an active MHEG-5 engine, exactly one Application object is active. The engine goes from an active to an idle state only by an execution of the Quit action; the semantics of this are explained in clause 10. The engine goes from an idle to an active state either

1. as a step in the execution of a Launch or a Spawn action; the semantics of this are explained in clause 10; or
2. as a result of the idle MHEG-5 engine being forced from the outside to activate an Application object.

The latter case is the topic of this subclause.

From the outside, an MHEG-5 engine is typically an application that runs on some device in a software environment. When this application is first activated, it shall get information from the outside where the first Application object is to be found. This information may come, for instance, from the user, from memory, or from the network, and should make it possible for the MHEG-5 engine to attach to the name space of the Application object, and to find the Application object itself.

Once the MHEG-5 engine has the information, it attempts to retrieve that Application object and to activate it. The latter is done by implicitly invoking the *Activation* behaviour of the object. In order to also activate a Scene (without which no user input or display is possible), the Application object should have a TransitionTo action encoded in its OnStartUp object; this action then activates the first Scene of the Application.

Annex D

(informative)

Definition of application domains

This annex specifies the features of this part of ISO/IEC 13522 that need to be defined by a specific MHEG-5 application domain. The goal of this annex is to provide a set of tools for creating an instance of an MHEG-5 engine and examples of table values.

D.1 Object interchange format

As an example, one specifies here that the application domain chooses the ASN.1 notation defined in Annex A for interchange format.

Interchange format: ASN.1.

D.2 Set of classes

The application domain defines a set of mandatory classes such as the following.

Action, Application, Audio, Bitmap, BooleanVariable, ContentRefVariable, EntryField, HotSpot, HyperText, IntegerVariable, InterchangedProgram, Link, ListGroup, ObjectRefVariable, OctetStringVariable, PushButton, Rectangle, RemoteProgram, ResidentProgram, Scene, Slider, Stream, SwitchButton., Text, TokenGroup, Video.

D.3 Set of features

The application domain defines a set of mandatory and optional features such as the following example.

Feature	Requirement
Ancillary connections	Mandatory
Caching	Optional
Cloning	Mandatory
Free moving cursor	Optional
Scaling (Video and Bitmap)	Optional
Stacking of Applications	Optional
Trick modes	Optional

D.4 Content data encoding

This part of ISO/IEC 13522 is generic in the sense that it does not specify exactly how the content data (e.g., bitmap data) is encoded. However, this part of ISO/IEC 13522 is also specific in the sense that

1. it specifies which type of content data is supported, and
2. it provides a method for defining - for a specific application domain - a list of the actual encoding used in that domain.

The following table exemplifies how an application domain could specify the formats needed, the values listed below are purely provided as an example of values an application domain could choose.

Content table:

Attribute	Permissible values
FontAttributes	bold, Italic, emphasis, strong emphasis

FontName	(None specified)
AbsoluteColour	RGB16, that is coding of graphics in 16 bits per pixel, allocating 5 bits to the Red and Blue components and 6 bits to the Green component.
CharacterSet	1 - referring to the subset of ISO/IEC 8859-1 as specified in HTML 2.0
TransitionEffect	(None specified)

Encoding table:

Type of content	Specification (Data Types)	Hook values
Font encoding format	(None specified)	
Palette encoding format	(None specified)	
Bitmap encoding format	2D graphics defined in DAVIC 1.0.	1
	MPEG-2 Video I-frames (ISO/IEC 13818-2)	2
Text encoding format	The subset of ISO/IEC 8859-1 as specified in HTML 2.0	1
EntryField encoding format	The subset of ISO/IEC 8859-1 as specified in HTML 2.0	1
HyperText encoding format	The subset of ISO/IEC 8859-1 as specified in HTML 2.0	1
Stream encoding format	MPEG-2 Systems specified in ISO/IEC 13818-1	1
	MPEG-1 Audio specified in ISO/IEC 11172-3	2
	AIFF-C	3
	Real time streams defined in DAVIC 1.0	4
LineArt encoding format	(None specified)	
CursorShape encoding format	(None specified)	
InterchangedProgram encoding format	MHEG-6	1

D.5 UserInput registers

In order to have a working instantiation standard, the application domain should specify one or more InputEventRegisters. Each register has a number, which is exchanged as one of the parameters of a Scene object. The content of an InputEventRegister (which is not exchanged) is a set of numbers (representing UserInputEventTags) and a name. It is recommended that the names have as much semantic content as possible («Up» is a good name; «13» is a bad name). The name/number pairs bind a specific UserInputEventTag to a logical input event. It is the task of the engine implementer to bind the logical input event to one or more physical input events.

The following table is an example of a set of InputEventRegisters:

Register #	UserInputEventTag	Name	Comment
1	1	Up	
	2	Down	
	3	Left	
	4	Right	
	5 - 14	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, respectively	
	15	Select	
	16	Exit	
	17	Help	
	18-99	Reserved for future specification.	
	100-	Vendor specific	

D.6 Semantic constraints on the MHEG-5 applications

An MHEG-5 application domain may constrain its applications in certain dimensions. The following table contains a list of features of this part of ISO/IEC 13522 that are optional or may be supported to a higher or lesser degree. For each of these points, an MHEG-5 application domain must make a decision, as outlined in the right-hand column.

The following table is an example of such decisions.

Feature	Constraint
FreeMovingCursor	Optional
ApplicationStacking	Optional
Scaling	Optional
SceneCoordinateSystem(X,Y)	The following coordinate systems are supported: 720x576, 704x576, 640x576, 544x576, 540x576, 480x576, 352x576, 352x288, 720x480, 704x480, 640x480, 544x480, 480x480, 352x480 and 352x240.
SceneAspectRatio(W,H)	The following aspect ratios are supported: 1/1, 4/3 and 16/9.
AncillaryConnections	Optional
TrickModes	Optional
MultipleRTGraphicsStreams(N)	One active RTGraphics at a time.
MultipleAudioStreams(N)	One audio stream of type <i>stream</i> and one of type <i>memory</i> at a time.
MultipleVideoStreams(N)	One video stream playing at a time.
OverlappingVisibles(N)	No constraint.
Cloning	Mandatory

D.7 EngineEvent

This application domain reserves no particular value of EngineEvent. This is left to the application developer.

D.8 GetEngineSupport

Permissible strings to submit in the GetEngineSupport action are the ones defined in this part of ISO/IEC 13522... In addition to these strings, a given application domain may define others; in which case, the application domain should define them precisely in a table similar to the one above. This example does not define any additional string.

D.9 Protocol mapping and external interaction

The following table provides examples to define mapping to external environment

MHEG-5 entity	Mapping needed.	Semantics of these MHEG-5 structures needs specification:
OpenConnection, CloseConnection	Mapping to connection management.	- In OpenConnection: - Protocol: one of the two strings: "PSTN" or "ISDN". - Address: E.164 NSAP.
RemoteProgram objects.	Mapping to RemoteProgram call protocol in the application domain.	- In Call and Fork: - Name - Parameters - ProgramConnectionTag Shall be consistent with DSM-CC
Application name space	Mapping to name space of the application domain.	- ObjectReference - ContentReference Shall be consistent with DSM-CC
Application name space in case a TransitionTo action uses the ConnectionTag parameter.	Mapping to the name space of the application domain	- ObjectReference - ContentReference Shall be consistent with DSM-CC
Persistent storage name space	Mapping to the name space of the persistent storage.	- In StorePersistent and ReadPersistent: - InFileName, OutFileName Shall be consistent with DSM-CC
Stream actions	Mapping to the stream interface of the application domain.	- In Stream - Speed - CounterPosition
Stream events	Mapping to stream states and stream events in the application domain.	- In Stream - StreamPlaying, StreamStopped (mapping to application-domain stream state machine) - CounterPosition - StreamEventTag

This page intentionally left blank

This page intentionally left blank

ICS 35.040

Descriptors: data processing, video data, audio data, graphic data processing, image processing, information interchange, communication procedure, coding (data conversion), coded representation.

Price based on 239 pages
