



EUROPEAN
TELECOMMUNICATION
STANDARD

FINAL DRAFT
pr **ETS 300 777-1**

June 1997

Source: ETSI EP-MTA

Reference: DE/MTA-001057-1

Formerly: DE/TE-01057-1

ICS: 33.020

Key words: API, MHEG, multimedia, terminal

**Terminal Equipment (TE);
End-to-end protocols for multimedia information
retrieval services;
Part 1: Coding of multimedia and hypermedia
information for basic
multimedia applications (MHEG-5)**

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE

Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE

X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 4 92 94 42 00 - Fax: +33 4 93 65 47 16

Copyright Notification: No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 1997. All rights reserved.

Contents

Foreword	5
1 Scope	7
2 Normative references	7
3 Definitions and abbreviations	7
3.1 Definitions	7
3.2 Abbreviations	7
4 MHEG-5 ASN.1 notation design principles	8
5 MHEG-5 ASN.1 notation	8
History.....	26

Blank page

Foreword

This final draft European Telecommunication Standard (ETS) has been produced by the Terminal Equipment (TE) Technical Committee of the European Telecommunications Standards Institute (ETSI), and is now submitted for the Voting phase of the ETSI standards approval procedure by its successor ETSI Project Multimedia Terminals and Applications (MTA).

This ETS consists of four parts as follows:

- Part 1:** "Coding of multimedia and hypermedia information for basic multimedia applications (MHEG-5)";
- Part 2: "Use of Digital Storage Media Command and Control (DSM-CC) for basic multimedia applications";
- Part 3: "Application Programmable Interface (API) for MHEG-5 ";
- Part 4: "Videotex Man Machine Interface (VEMMI) enhancements to support broadband multimedia information retrieval services".

Proposed transposition dates	
Date of latest announcement of this ETS (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

Blank page

1 Scope

This European Telecommunications Standard (ETS) specifies the MHEG part 5 Abstract Syntax Notation One (ASN.1) notation consisting of a syntax description (equivalent to the Extended Backus Naur Form (EBNF) syntax) and encoding rules.

MHEG part 5 (ISO/IEC IS 13522-5 [1]) specifies the coded representation of interchanged multimedia/hypermedia information objects (MHEG-5 objects) for use in the domain of base-level interactive applications such as movies-on-demand, teleshopping, near video-on-demand.

MHEG-5 specifies objects and their semantics using an informal text description. It also provides a formal description of the interchanged objects syntax using EBNF.

This specification is included in ISO/IEC IS 13522-5 [1] as normative annex A.

2 Normative references

This ETS incorporates by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

- [1] ISO/IEC 13522-5 (1996): "Information technology - Coding of Multimedia and Hypermedia information - Part 5: Support for Base-Level Interactive Applications".
- [2] ISO/IEC 8824-1 (1996)/ITU-T Recommendation X.680 (1995): "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation - Amendment 1".
- [3] ISO/IEC 8825-1 (1996)/ITU-T Recommendation X.690 (1995): "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of this ETS, the definitions in ISO/IEC IS 13522-5 [1], ISO/IEC 8824-1 [2] and ISO/IEC 8825-1 [3] apply.

3.2 Abbreviations

For the purposes of this ETS, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
DER	Distinguished Encoding Rules
EBNF	Extended Backus Naur Form
MHEG	Multimedia and Hypermedia Expert Group

4 MHEG-5 ASN.1 notation design principles

Designing an ASN.1 encoding is always a trade-off between the compactness of the encoded datasyntax and the compactness of the ASN.1 parser required. As this syntax is designed to run on a minimal resource platform, the compactness of the ASN.1 parser is favoured with regard to the compactness of the interchanged code.

These constraints lead to the following design principles:

- 1) use of semantic constraints to allow parsers to do more conformance checking;
- 2) use of context-free tags to allow parsers to do strong type checking;
- 3) uniform structure for elementary action to allow parsers to use a general algorithm.

5 MHEG-5 ASN.1 notation

This clause defines the ASN.1 notation for the syntax of MHEG-5 objects conforming to ISO/IEC 13522-5 [1].

The encoding of the MHEG-5 objects from this ASN.1 syntax shall make use of the Distinguished Encoding Rules (DER) defined in ISO/IEC 8825-1 [3].

The syntax that shall be used in the DER is detailed below:

```
--$PREFIX=ISOMHEG-mheg-5:mheg-5
-- Module: mheg-5

ISO13522-MHEG-5
{joint-iso-itu-t(2) mheg(19) version(1) mheg-5(17)} DEFINITIONS IMPLICIT
TAGS ::=
BEGIN

-- This module defines the MHEG-5 abstract syntax which consists of data values of type:
-- ISO13522-MHEG-5.InterchangedObject.
-- This abstract syntax is identified by the name: {joint-iso-itu-t(2) mheg(19) version(1)
mheg-5(17)}.

InterchangedObject ::= CHOICE
{
  application    [0] ApplicationClass,
  scene          [1] SceneClass
}

-- Root Class _____
RootClass ::= ObjectReference

-- Group Class _____
GroupClass ::= SET
{
  RootClass (WITH COMPONENTS
    {external-reference (WITH COMPONENTS {..., object-number (0)}) PRESENT,
    internal-reference ABSENT}),
  standard-identifier [2] StandardIdentifier OPTIONAL,
  standard-version [3] INTEGER (1) OPTIONAL,
  object-information [4] OCTET STRING OPTIONAL,
  on-start-up [5] ActionClass OPTIONAL,
  on-close-down [6] ActionClass OPTIONAL,
  original-group-cache-priority [7] INTEGER (0..255) DEFAULT 127,
  items [8] SEQUENCE SIZE (1..MAX) OF GroupItem OPTIONAL
}

StandardIdentifier ::= SEQUENCE
{
  joint-iso-itu INTEGER (2),
  mheg INTEGER (19)
}

GroupItem ::= CHOICE
{
  resident-program [9] ResidentProgramClass,
  remote-program [10] RemoteProgramClass,
  interchanged-program [11] InterchangedProgramClass,
  palette [12] PaletteClass,
  font [13] FontClass,
  cursor-shape [14] CursorShapeClass,
```



```

boolean-variable [15] BooleanVariableClass,
integer-variable [16] IntegerVariableClass,
octet-string-variable [17] OctetStringVariableClass,
object-ref-variable [18] ObjectRefVariableClass,
content-ref-variable [19] ContentRefVariableClass,
link [20] LinkClass,
stream [21] StreamClass,
bitmap [22] BitmapClass,
line-art [23] LineArtClass,
dynamic-line-art [24] DynamicLineArtClass,
rectangle [25] RectangleClass,
hotspot [26] HotspotClass,
switch-button [27] SwitchButtonClass,
push-button [28] PushButtonClass,
text [29] TextClass,
entry-field [30] EntryFieldClass,
hyper-text [31] HyperTextClass,
slider [32] SliderClass,
token-group [33] TokenGroupClass,
list-group [34] ListGroupClass
}

-- . Application Class _____

ApplicationClass ::= SET
{
  COMPONENTS OF GroupClass,
  on-spawn-close-down [35] ActionClass OPTIONAL,
  on-restart [36] ActionClass OPTIONAL,
  default-attributes [37] SEQUENCE SIZE (1..MAX) OF DefaultAttribute OPTIONAL
}

DefaultAttribute ::= CHOICE
{
  character-set [38] INTEGER,
  background-colour [39] Colour,
  text-content-hook [40] INTEGER,
  text-colour [41] Colour,
  font [42] FontBody,
  font-attributes [43] OCTET STRING,
  interchanged-program-content-hook [44] INTEGER,
  stream-content-hook [45] INTEGER,
  bitmap-content-hook [46] INTEGER,
  line-art-content-hook [47] INTEGER,
  button-ref-colour [48] Colour,
  highlight-ref-colour [49] Colour,
  slider-ref-colour [50] Colour
}

FontBody ::= CHOICE
{
  direct-font OCTET STRING,
  indirect-font ObjectReference
}

-- Scene Class _____

SceneClass ::= SET
{
  COMPONENTS OF GroupClass,
  input-event-register [51] INTEGER,
  scene-coordinate-system [52] SceneCoordinateSystem,
  aspect-ratio [53] AspectRatio DEFAULT {width 4, height 3},
  moving-cursor [54] BOOLEAN DEFAULT FALSE,
  next-scenes [55] SEQUENCE SIZE (1..MAX) OF NextScene OPTIONAL
}

SceneCoordinateSystem ::= SEQUENCE
{
  x-scene INTEGER,
  y-scene INTEGER
}

AspectRatio ::= SEQUENCE
{
  width INTEGER,
  height INTEGER
}

NextScene ::= SEQUENCE
{
  scene-ref OCTET STRING,
  scene-weight INTEGER (0..255)
}

-- . Ingredient Class _____

```

```
IngredientClass ::= SET
{
  RootClass (WITH COMPONENTS
    {..., external-reference (WITH COMPONENTS {..., object-number (1..MAX))}),
    initially-active [56] BOOLEAN DEFAULT TRUE,
    content-hook [57] INTEGER OPTIONAL,
    original-content [58] ContentBody OPTIONAL,
    shared [59] BOOLEAN DEFAULT FALSE
  )
}

ContentBody ::= CHOICE
{
  included-content OCTET STRING,
  referenced-content ReferencedContent
}

ReferencedContent ::= SEQUENCE
{
  content-reference ContentReference,
  content-size [60] INTEGER OPTIONAL,
  content-cache-priority [61] INTEGER (0..255) DEFAULT 127
}

-- . Link Class _____

LinkClass ::= SET
{
  COMPONENTS OF IngredientClass
  (WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT}),
  link-condition [62] LinkCondition,
  link-effect [63] ActionClass
}

LinkCondition ::= SEQUENCE
{
  event-source ObjectReference,
  event-type EventType,
  event-data EventData OPTIONAL
}

EventType ::= ENUMERATED
{
  is-available(1),
  content-available(2),
  is-deleted(3),
  is-running(4),
  is-stopped(5),
  user-input(6),
  anchor-fired(7),
  timer-fired(8),
  asynch-stopped(9),
  interaction-completed(10),
  token-moved-from(11),
  token-moved-to(12),
  stream-event(13),
  stream-playing(14),
  stream-stopped(15),
  counter-trigger(16),
  highlight-on(17),
  highlight-off(18),
  cursor-enter(19),
  cursor-leave(20),
  is-selected(21),
  is-deselected(22),
  test-event(23),
  first-item-presented(24),
  last-item-presented(25),
  head-items(26),
  tail-items(27),
  item-selected(28),
  item-deselected(29),
  entry-field-full(30),
  engine-event(31)
}

EventData ::= CHOICE
{
  octetstring OCTET STRING,
  boolean BOOLEAN,
  integer INTEGER
}

-- Program Class _____
```

```

ProgramClass ::= SET
{
  COMPONENTS OF IngredientClass
    (WITH COMPONENTS {..., initially-active (FALSE) PRESENT}),
  name [64] OCTET STRING,
  initially-available [65] BOOLEAN DEFAULT TRUE
}

-- Resident Program Class _____

ResidentProgramClass ::= ProgramClass
(WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT})

-- Remote Program Class _____

RemoteProgramClass ::= SET
{
  COMPONENTS OF ProgramClass
    (WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT}),
  program-connection-tag [66] INTEGER OPTIONAL
}

-- Interchanged Program Class _____

InterchangedProgramClass ::= ProgramClass
(WITH COMPONENTS {..., original-content PRESENT})

-- Palette Class _____

PaletteClass ::= IngredientClass
(WITH COMPONENTS
  {..., content-hook PRESENT, original-content PRESENT, initially-active (TRUE)})

-- Font Class _____

FontClass ::= IngredientClass
(WITH COMPONENTS
  {..., content-hook PRESENT, original-content PRESENT, initially-active (TRUE)})

-- Cursor Shape _____

CursorShapeClass ::= IngredientClass
(WITH COMPONENTS
  {..., content-hook PRESENT, original-content PRESENT, initially-active (TRUE)})

-- Variable Class _____

VariableClass ::= SET
{
  COMPONENTS OF IngredientClass
    (WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT, initially-active
    (TRUE)}),
  original-value [67] OriginalValue
}

OriginalValue ::= CHOICE
{
  boolean BOOLEAN,
  integer INTEGER,
  octetstring OCTET STRING,
  object-reference [68] ObjectReference,
  content-reference [69] ContentReference
}

-- Boolean Variable Class _____

BooleanVariableClass ::= VariableClass
(WITH COMPONENTS {..., original-value (WITH COMPONENTS {..., boolean PRESENT})})

-- Integer Variable Class _____

IntegerVariableClass ::= VariableClass
(WITH COMPONENTS {..., original-value (WITH COMPONENTS {..., integer PRESENT})})

-- Octet String Variable Class _____

OctetStringVariableClass ::= VariableClass
(WITH COMPONENTS {..., original-value (WITH COMPONENTS {..., octetstring PRESENT})})

-- Object Reference Variable Class _____

ObjectRefVariableClass ::= VariableClass
(WITH COMPONENTS {..., original-value (WITH COMPONENTS {..., object-reference PRESENT})})

-- Content Reference Variable Class _____

```

```
ContentRefVariableClass ::= VariableClass
  (WITH COMPONENTS {..., original-value (WITH COMPONENTS {..., content-reference PRESENT}))})

-- Presentable Class _____
PresentableClass ::= IngredientClass

-- Token Manager Class _____
TokenManagerClass ::= SET
{
  movement-table [70] SEQUENCE SIZE (1..MAX) OF Movement OPTIONAL
}

Movement ::= SEQUENCE SIZE (1..MAX) OF INTEGER

-- Token Group Class _____
TokenGroupClass ::= SET
{
  COMPONENTS OF PresentableClass
  (WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT}),
  COMPONENTS OF TokenManagerClass,
  token-group-items [71] SEQUENCE SIZE (1..MAX) OF TokenGroupItem,
  no-token-action-slots [72] SEQUENCE SIZE (1..MAX) OF ActionSlot OPTIONAL
}

TokenGroupItem ::= SEQUENCE
{
  a-visible ObjectReference,
  action-slots SEQUENCE SIZE (1..MAX) OF ActionSlot OPTIONAL
}

ActionSlot ::= CHOICE
{
  action-class ActionClass,
  null NULL
}

-- List Group Class _____
ListGroupClass ::= SET
{
  COMPONENTS OF TokenGroupClass,
  positions [73] SEQUENCE SIZE (1..MAX) OF XYPosition,
  wrap-around [74] BOOLEAN DEFAULT FALSE,
  multiple-selection [75] BOOLEAN DEFAULT FALSE
}

-- Visible Class _____
VisibleClass ::= SET
{
  COMPONENTS OF PresentableClass,
  original-box-size [76] OriginalBoxSize,
  original-position [77] XYPosition DEFAULT {x-position 0, y-position 0},
  original-palette-ref [78] ObjectReference OPTIONAL
}

OriginalBoxSize ::= SEQUENCE
{
  x-length INTEGER (0..MAX),
  y-length INTEGER (0..MAX)
}

-- Bitmap Class _____
BitmapClass ::= SET
{
  COMPONENTS OF VisibleClass
  (WITH COMPONENTS {..., original-content PRESENT}),
  tiling [79] BOOLEAN DEFAULT FALSE,
  original-transparency [80] INTEGER (0..100) DEFAULT 0
}

-- Line Art Class _____
LineArtClass ::= SET
{
  COMPONENTS OF VisibleClass
  (WITH COMPONENTS {..., original-content PRESENT}),
  bordered-bounding-box [81] BOOLEAN DEFAULT TRUE,
  original-line-width [82] INTEGER DEFAULT 1,
  original-line-style [83] INTEGER {solid(1), dashed(2), dotted(3)} DEFAULT solid,
}
```

```
    original-ref-line-colour [84] Colour OPTIONAL,  
    original-ref-fill-colour [85] Colour OPTIONAL  
}  
  
-- Rectangle Class _____  
RectangleClass ::= LineArtClass  
  (WITH COMPONENTS  
   {..., content-hook ABSENT, original-content ABSENT, bordered-bounding-box ABSENT})  
  
-- Dynamic Line Art Class _____  
DynamicLineArtClass ::= LineArtClass  
  (WITH COMPONENTS  
   {..., content-hook ABSENT, original-content ABSENT})  
  
-- Text Class _____  
TextClass ::= SET  
{  
  COMPONENTS OF VisibleClass  
  (WITH COMPONENTS {..., original-content PRESENT}),  
  original-font [86] FontBody OPTIONAL,  
  font-attributes [43] OCTET STRING OPTIONAL,  
  text-colour [41] Colour OPTIONAL,  
  background-colour [39] Colour OPTIONAL,  
  character-set [38] INTEGER OPTIONAL,  
  horizontal-justification [87] Justification DEFAULT start,  
  vertical-justification [88] Justification DEFAULT start,  
  line-orientation [89] LineOrientation DEFAULT horizontal,  
  start-corner [90] StartCorner DEFAULT upper-left,  
  text-wrapping [91] BOOLEAN DEFAULT FALSE  
}  
  
Justification ::= ENUMERATED  
{  
  start(1),  
  end(2),  
  centre(3),  
  justified(4)  
}  
  
LineOrientation ::= ENUMERATED {vertical(1), horizontal(2)}  
  
StartCorner ::= ENUMERATED  
{  
  upper-left(1),  
  upper-right(2),  
  lower-left(3),  
  lower-right(4)  
}  
  
-- Stream Class _____  
StreamClass ::= SET  
{  
  COMPONENTS OF PresentableClass  
  (WITH COMPONENTS {..., original-content PRESENT}),  
  multiplex [92] SEQUENCE SIZE (1..MAX) OF StreamComponent,  
  storage [93] Storage DEFAULT stream,  
  looping [94] INTEGER {infinity(0)} DEFAULT 1  
}  
  
StreamComponent ::= CHOICE  
{  
  audio [95] AudioClass,  
  video [96] VideoClass,  
  rtgraphics [97] RTGraphicsClass  
}  
  
Storage ::= ENUMERATED {memory(1), stream(2)}  
  
-- Audio Class _____  
AudioClass ::= SET  
{  
  COMPONENTS OF PresentableClass  
  (WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT, shared ABSENT}),  
  component-tag [98] INTEGER,  
  original-volume [99] INTEGER DEFAULT 0  
}  
  
-- Video Class _____  
VideoClass ::= SET
```

```

{
  COMPONENTS OF VisibleClass
  (WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT, shared ABSENT, original-
palette-ref ABSENT}),
  component-tag      [98] INTEGER,
  termination       [100] Termination DEFAULT disappear
}

Termination ::= ENUMERATED {freeze(1), disappear(2)}

-- RTGraphics Class _____
RTGraphicsClass ::= SET
{
  COMPONENTS OF VisibleClass
  (WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT, shared ABSENT}),
  component-tag      [98] INTEGER,
  termination       [100] Termination DEFAULT disappear
}

-- Interactable Class _____
InteractableClass ::= SET
{
  engine-resp       [101] BOOLEAN DEFAULT TRUE,
  highlight-ref-colour [49] Colour OPTIONAL
}

-- Slider Class _____
SliderClass ::= SET
{
  COMPONENTS OF VisibleClass
  (WITH COMPONENTS {..., content-hook ABSENT, original-content ABSENT}),
  COMPONENTS OF InteractableClass,
  orientation       [102] Orientation,
  max-value         [103] INTEGER,
  min-value         [104] INTEGER DEFAULT 1,
  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)}

-- 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)}

-- Hyper Text Class _____
HyperTextClass ::= SET
{
  COMPONENTS OF TextClass,
  COMPONENTS OF InteractableClass
}

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

-- Hotspot Class _____
HotspotClass ::= ButtonClass

```

```

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

-- Switch Button Class _____
SwitchButtonClass ::= SET
{
  COMPONENTS OF PushButtonClass,
  button-style [114] ButtonStyle
}

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

-- 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] GetFillColour,
  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] SetFillColour,
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] SetLineColour,
set-line-style [202] SetLineStyle,
set-line-width [203] SetLineWidth,
set-override-mode [204] SetOverrideMode,
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
{
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  
}  
  
SetFillColour ::= 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
{
  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
}

-- 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
```

History

Document history			
July 1996	Public Enquiry	PE 110:	1996-07-22 to 1996-11-15
June 1997	Vote	V 9735:	1997-06-17 to 1997-08-29