Terminal Equipment (TE);
End-to-end protocols for, multimedia information retrieval services;
Part 3: Application Programmable Interface (API) for MHEG-5
## Contents

Foreword .................................................................................................................................................. 5  
1 Scope .................................................................................................................................................. 7  
2 Normative references .......................................................................................................................... 7  
3 Definitions and abbreviations ........................................................................................................... 7  
3.1 Definitions ....................................................................................................................................... 7  
3.2 Abbreviations ................................................................................................................................. 8  
4 Overview ............................................................................................................................................ 8  
4.1 The DAVIC application interchange format ..................................................................................... 8  
4.2 Core set of Java APIs ...................................................................................................................... 8  
5 The MHEG-5 API ............................................................................................................................... 9  
6 Map between MHEG-5 elementary actions and MHEG-5 API operations ........................................ 26  
History .................................................................................................................................................... 29
Foreword

This European Telecommunication Standard (ETS) has been produced by the Terminal Equipment (TE) Technical Committee of the European Telecommunications Standards Institute (ETSI).

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".

<table>
<thead>
<tr>
<th>Transposition dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of adoption:</td>
</tr>
<tr>
<td>Date of latest announcement of this ETS (doa):</td>
</tr>
<tr>
<td>Date of latest publication of new National Standard or endorsement of this ETS (dop/e):</td>
</tr>
<tr>
<td>Date of withdrawal of any conflicting National Standard (dow):</td>
</tr>
</tbody>
</table>
1 Scope

This European Telecommunications Standard (ETS) specifies the Application Programmable Interface (API) for the manipulation of multimedia and hypermedia information objects, i.e. the API that shall be provided by MHEG-5 engines for their control by applications running on a DAVIC 1.1 compliant terminal.

MHEG part 5 (ISO/IEC 13522-5 [1]) is a standard, which specifies the coded representation of interchanged multimedia/hypermedia information objects (MHEG-5 objects) for base level applications. These so-called MHEG-5 objects are handled, interpreted and presented by MHEG-5 engines.

2 Normative references

This ETS incorporates by dated and undated reference, provisions from other publications. These 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.


[2] ETS 300 777-1: "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)".

[3] ETS 300 777-2: "Terminal Equipment (TE); End-to-end protocols for multimedia information retrieval services; Part 2: Use of Digital Storage Media Command and Control (DSM-CC) for basic multimedia applications".


3 Definitions and abbreviations

3.1 Definitions

For the purposes of this ETS, the definitions of the standards referenced below apply. Should any ambiguity occur, definitions of the following standards apply, in decreasing order:

- any other standard part of ISO/IEC 13522 MHEG.

Application Programmable Interface (API): A boundary across which a software application uses facilities of programming languages to invoke software services. These facilities may include procedures or operations, shared data objects and resolution of identifiers.

local application: A piece of software which is part of the (telecommunication) application and is running on the considered equipment.

MHEG-5 API: The API provided by an MHEG-5 engine to local applications for the manipulation of MHEG-5 objects, as defined in this ETS.

MHEG-5 engine: A process or a set of processes that interpret MHEG-5 objects encoded according to the encoding specifications of ETS 300 777-1 [2] or the MHEG-5 textual notation.
3.2 Abbreviations

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

API  Application Programmable Interface
ASN.1     Abstract Syntax Notation One
DAVIC     Digital Audio Visual Council
DSM-CC    Digital Storage Media Command and Control
MHEG      Multimedia and Hypermedia information coding Experts Group
SI        Service Information
STU       Set Top Unit
U-U       User to User
VM        Virtual Machine

4 Overview

The following subclause positions the API defined by this ETS in the framework of the DAVIC specifications.

4.1 The DAVIC application interchange format

To deliver multimedia information to STUs in an interoperable way, applications shall use the MHEG-5 final form interchange format, as defined by ISO/IEC 13522-5 [1]. The ASN.1 notation and encoding, as defined by ETS 300 777-1 [2], shall be used to interchange MHEG-5 objects. This format defines the semantics and the encoding of the multimedia and hypermedia objects.

To deliver program code to STUs in an interoperable way, applications shall use the MHEG-5 Interchanged Program class to encapsulate DAVIC VM code, according to the semantics and encoding defined by ISO/IEC 13522-6 [4]. Java VM classes are called from MHEG-5 objects using the MHEG-5 call and fork elementary actions.

The Java VM code interchange unit is a Java VM class. Java VM classes shall be encoded as defined by the Class File Format section of the Java Virtual Machine specification. A Java class encapsulates data and methods that consist of sequences of instructions. The instruction set is defined by the Java Virtual Machine instruction set section of the Java Virtual Machine specification.

4.2 Core set of Java APIs

The following set of APIs are used by Java VM code in the DAVIC 1.1 specifications to express access to basic functions of the STU in an interoperable way:

- the java.lang package;
- the java.util package;
- the iso.mheg5 package;
- the davic.dsmccuu package;
- the etsi.si package.

NOTE 1: The Java VM specification provides flexible mechanisms to call upon external functions whose interface is defined as a Java package. The DAVIC 1.1 specification only includes a minimum core set of packages required for Java VM code to be useful in a DAVIC environment. It is anticipated that additional Java packages will be standardized at a later stage.

NOTE 2: Especially, the java.io package, although strictly speaking not necessary to the useful performance of the VM environment, is part of the Java foundation classes. It is intended that the java.io package be added to the DAVIC core set of Java APIs together with an adequate specification of its semantics in a DAVIC environment.

The java.lang package, as defined by the Java API documentation, consists of the minimal set of Java VM classes needed to run Java VM code, supporting the following functionality: basic data types, object, mathematical operations, security, thread management, string manipulation, exception handling.
The `java.util` package, as defined by the Java API documentation, consists of Java VM classes supporting a number of utility features common to all Java VM programs.

The `iso.mheg5` package, as defined by this ETS, provides Java VM code with access to and manipulation of the MHEG-5 multimedia presentation and interaction objects, i.e. access to the dynamic attributes of MHEG-5 objects and invocation of elementary actions on MHEG-5 objects.

The `davic.dsmccuu` package, together with the associated `davic.CosNaming` and `davic.CosNaming.NamingContext_` packages, as defined by ETS 300 777-2 [3], enables Java VM code to use the DSM-CC U-U interface objects for network data access.

The `davic.dsmccuu` package implements a subset of the DSM-CC U-U API. Access to the following Core SetTop services is provided:

- interface Base: operations Close and Destroy;
- interface File: operations Read and Write;
- interface Directory: operations Open, Close and Get;
- interface ServiceGateway: operations Attach and Detach;
- interface CosNaming::NamingContext: operations List and Resolve;
- interface CosNaming::BindingIterator: operations Next_One and Next_N.

The `etsi.si` package enables Java VM code to access information transmitted in the DAVIC Service Information (SI) stream.

5 The MHEG-5 API

```java
// Package
package iso.mheg5;

// Useful definitions
public class ObjectReference {
    /* Attributes */
    public byte[] groupIdentifier;
    public int objectNumber;
    /* Constructors */
    public ObjectReference();
    public ObjectReference(int objectNumber);
    public ObjectReference(byte[] groupIdentifier, int objectNumber);
}

public class ContentReference {
    /* Attributes */
    byte[] reference;
    /* Constructors */
    public ContentReference();
    public ContentReference(byte[] reference);
}

public class MhegException extends java.lang.Exception {
```
/* Attributes */
// Constant declarations for exceptionCode
public static final short TARGET_NOT_AVAILABLE = 1;
public static final short INVALID_TARGET = 2;
public static final short INVALID_PARAMETER = 3;

public short exceptionCode;
public short parameterRank;

/* Constructors */
protected MhegException();

// Construct an MhegException with exceptionCode identified by the reason parameter
public MhegException(
short reason)
{
    exceptionCode = reason;
    parameterRank = -1;
}

// Construct an MhegException with exceptionCode identified by the reason parameter and parameterRank
// by the position parameter
public MhegException(
short reason,
short position)
{
    exceptionCode = reason;
    parameterRank = position;
}

abstract public class Root
{

    /* Constructors */
    protected Root();

    /* Methods */

    // Return the reference of the Java object associated with the MHEG-5 object whose identification is
    // mheg5ObjectReference
    // If the Java object does not exist, create it first
    public static final Root getObject(
        ObjectReference mheg5ObjectReference)
    throws MhegException;

    // Correspond to the GetAvailabilityStatus MHEG-5 elementary action
    public Boolean getAvailabilityStatus()
    throws MhegException;

    // Correspond to the GetRunningStatus MHEG-5 elementary action
    public Boolean getRunningStatus()
    throws MhegException;

}

abstract public class Group extends Root
{

    /* Constructors */
    protected Group();

    /* Methods */

    // Correspond to the SetCachePriority MHEG-5 elementary action
    // The cachePriority parameter value shall be within the range [0,255]
    public void setCachePriority(
        byte cachePriority)
    throws MhegException;

    // Retrieve the value of the GroupCachePriority attribute
    public Integer getCachePriority()
    throws MhegException;

}
public class Application extends Group
{
/* Constructors */
protected Application();
/* Methods */
// Correspond to the LockScreen MHEG-5 elementary action
public void lockScreen() throws MhegException;
// Correspond to the UnlockScreen MHEG-5 elementary action
public void unlockScreen() throws MhegException;
// Retrieve the value of the LockCount attribute
// The wrapped returned value shall be positive or equal to zero
public Integer getLockCount() throws MhegException;
// Correspond to the GetEngineSupport MHEG-5 elementary action
public Boolean getEngineSupport(byte[] feature) throws MhegException;
}

public class Scene extends Group
{
/* Constructors */
protected Scene();
/* Methods */
// Correspond to the TransitionTo MHEG-5 elementary action
public void transitionTo() throws MhegException;
public void transitionTo(int connectionTag) throws MhegException;
public void transitionTo(byte transitionEffect) throws MhegException;
public void transitionTo(int connectionTag, byte transitionEffect) throws MhegException;
// Correspond to the SetTimer MHEG-5 elementary action
public void setTimer(int timerId) throws MhegException;
public void setTimer(int timerId, int timerValue) throws MhegException;
public void setTimer(int timerId, int timerValue, boolean absoluteTime) throws MhegException;
// Retrieve the value of the TimerPosition (in the timerPosition parameter) and AbsoluteTime (in the absoluteTime parameter) fields of the timer whose identification is timerId in the Timers attribute
// If the targeted timer exists, the method returns true otherwise false
public boolean getTimer(int timerId, Integer timerPosition, Boolean absoluteTime) throws MhegException;
// Correspond to the SendEvent MHEG-5 elementary action
public void sendEvent(ObjectReference eventSource, byte eventType) throws MhegException;
public void sendEvent(
ObjectReference eventSource,
byte eventType,
boolean eventData)
throws MhegException;
public void sendEvent(
ObjectReference eventSource,
byte eventType,
int eventData)
throws MhegException;
public void sendEvent(
ObjectReference eventSource,
byte eventType,
byte[] eventData)
throws MhegException;

// Correspond to the SetCursorShape MHEG-5 elementary action
public void setCursorShape()
throws MhegException;
public void setCursorShape(
ObjectReference cursorShape)
throws MhegException;

// Retrieve the shape of the cursor
// A returned null object reference indicates that the cursor has been removed from the scene
public ObjectReference getCursorShape()
throws MhegException;

// Correspond to the SetCursorPosition MHEG-5 elementary action
public void setCursorPosition(
short xCursor,
short yCursor)
throws MhegException;

// Correspond to the GetCursorPosition MHEG-5 elementary action
public void getCursorPosition(
Integer xCursor,
Integer yCursor)
throws MhegException;

abstract public class Ingredient extends Root
{

    /* Constructors */
    protected Ingredient();

    /* Methods */
    public void setData(
byte[] includedContent)
throws MhegException;
public void setData(
ContentReference referencedContent)
throws MhegException;
public void setData(
ContentReference referencedContent,
int contentSize)
throws MhegException;
public void setData(
ContentReference referencedContent,
byte contentCachePriority)
throws MhegException;
public void setData(
ContentReference referencedContent,
int contentSize,
byte contentCachePriority)
throws MhegException;

    // Retrieve the value of the ContentData attribute
    // The returned value shall be true and content shall be of type byte[] for an included content
    // The returned value shall be false and content shall be of type ContentReference for a referenced content
    public boolean getData(
Object content,
Integer contentSize,
Integer contentCachePriority)
throws MhegException;

    // Correspond to the Clone MHEG-5 elementary action
public ObjectReference clone()
    throws MhegException;

// Correspond to the Preload MHEG-5 elementary action
public void preload()
    throws MhegException;

// Correspond to the Unload MHEG-5 elementary action
public void unload()
    throws MhegException;

}  

// -----------------------------------------------------------------------------

public class Link extends Ingredient  
{  
/* Constructors */  
    protected Link();

/* Methods */
// Correspond to the Activate MHEG-5 elementary action
public void activate()
    throws MhegException;

// Correspond to the Deactivate MHEG-5 elementary action
public void deivate()
    throws MhegException;

}  

// -----------------------------------------------------------------------------

public class Program extends Ingredient  
{  
/* Constructors */  
    protected Program();

/* Methods */  
// Correspond to the Call MHEG-5 elementary action
// Every element of the parameters array shall be of one of the following types: Boolean, Integer, byte[], ObjectReference or ContentReference; otherwise, an exception is raised
public boolean call(
    Object[] parameters)
    throws MhegException;

// Correspond to the Fork MHEG-5 elementary action
// Every element of the parameters array shall be of one of the following types: Boolean, Integer, byte[], ObjectReference or ContentReference; otherwise, an exception is raised
public boolean fork(
    Object[] parameters)
    throws MhegException;

// Correspond to the Stop MHEG-5 elementary action
public void stop()
    throws MhegException;

}  

// -----------------------------------------------------------------------------

public class ResidentProgram extends Program  
{  
/* Constructors */
    protected ResidentProgram();

}  

// -----------------------------------------------------------------------------

public class RemoteProgram extends Program  
{  
/* Constructors */
    protected RemoteProgram();
public class InterchangedProgram extends Program
{
    /* Constructors */
    protected InterchangedProgram();
}

public class Palette extends Ingredient
{
    /* Constructors */
    protected Palette();
}

public class Font extends Ingredient
{
    /* Constructors */
    protected Font();
}

public class CursorShape extends Ingredient
{
    /* Constructors */
    protected CursorShape();
}

public class Variable extends Ingredient
{
    /* Constructors */
    protected Variable();
}

public class BooleanVariable extends Variable
{
    /* Constructors */
    protected BooleanVariable();
    /* Methods */
    // Correspond to the SetVariable MHEG-5 elementary action targeted at a BooleanVariable object
    public void setVariable(
        boolean value)
        throws MhegException;
    // Retrieve the value of the variable
    public Boolean getVariable()
        throws MhegException;
}

public class IntegerVariable extends Variable
{
    /* Constructors */
    protected IntegerVariable();
    /* Methods */
    // Correspond to the SetVariable MHEG-5 elementary action targeted at an IntegerVariable object
    public void setVariable(
public class OctetStringVariable extends Variable {
    /* Constructors */
    protected OctetStringVariable();
    /* Methods */
    // Correspond to the SetVariable MHEG-5 elementary action targetted at an OctetStringVariable object
    public void setVariable(  
        byte[] value)  
        throws MhegException;
    // Retrieve the value of the variable
    public byte[] getVariable()  
        throws MhegException;
    // Correspond to the Append MHEG-5 elementary action
    public void append(  
        byte[] value)  
        throws MhegException;
}

/* Constructors */
protected ObjectRefVariable();
/* Methods */
// Correspond to the SetVariable MHEG-5 elementary action targetted at an ObjectRefVariable object
public void setVariable( 
    ObjectReference value) 
    throws MhegException;
// Retrieve the value of the variable
public ObjectReference getVariable() 
    throws MhegException;
public class ContentRefVariable extends Variable
{

/* Constructors */
    protected ContentRefVariable();

/* Methods */
    // Correspond to the SetVariable MHEG-5 elementary action targeted at an ContentRefVariable object
    public void setVariable(
            ContentReference value)
        throws MhegException;

    // Retrieve the value of the variable
    public ContentReference getVariable()
        throws MhegException;
}

abstract public class Presentable extends Ingredient
{

/* Constructors */
    protected Presentable();

/* Methods */
    // Correspond to the Run MHEG-5 elementary action
    public void run()
        throws MhegException;

    // Correspond to the Stop MHEG-5 elementary action
    public void stop()
        throws MhegException;
}

public interface TokenManager
{

/* Methods */
    // Correspond to the Move MHEG-5 elementary action
    public void move(
            short movementId)
        throws MhegException;

    // Correspond to the MoveTo MHEG-5 elementary action
    // The index parameter value shall be within the range [0, number of elements in the group]
    public void moveTo(
            short index)
        throws MhegException;

    // Correspond to the GetTokenPosition MHEG-5 elementary action
    public Integer getTokenPosition()
        throws MhegException;
}

public class TokenGroup extends Presentable implements TokenManager
{

/* Constructors */
    protected TokenGroup();

/* Methods */
    // Correspond to the implementation of the TokenManager.move method
    public void move(
            short movementId)
        throws MhegException;

    // Correspond to the implementation of the TokenManager.moveTo method
    // The index parameter value shall be within the range [0, number of elements in the group]
    public void moveTo(
            short index)
        throws MhegException;

    // Correspond to the implementation of the TokenManager.getTokenPosition method

public Integer getTokenPosition()  
    throws MhegException;

    // Correspond to the CallActionSlot MHEG-5 elementary action
    // The index parameter value shall be within the range [0, number of elements in the group]
    public void callActionSlot(  
        short index  
    )  
    throws MhegException;

// -----------------------------------------------------------------------------
public class ListGroup extends TokenGroup
{
    /* Constructors */
    protected ListGroup();

    /* Methods */
    // Correspond to the AddItem MHEG-5 elementary action
    public void addItem(  
        short itemIndex,  
        ObjectReference visibleReference)  
    throws MhegException;

    // Correspond to the DelItem MHEG-5 elementary action
    public void delItem(  
        ObjectReference visibleReference)  
    throws MhegException;

    // Correspond to the GetListItem MHEG-5 elementary action
    public ObjectReference getListItem(  
        short itemIndex)  
    throws MhegException;

    // Correspond to the GetCellItem MHEG-5 elementary action
    public ObjectReference getCellItem(  
        short cellIndex)  
    throws MhegException;

    // Correspond to the GetItemStatus MHEG-5 elementary action
    public Boolean getItemStatus(  
        short itemIndex)  
    throws MhegException;

    // Correspond to the SelectItem MHEG-5 elementary action
    public void selectItem(  
        short itemIndex)  
    throws MhegException;

    // Correspond to the DeselectItem MHEG-5 elementary action
    public void deselectItem(  
        short itemIndex)  
    throws MhegException;

    // Correspond to the ToggleItem MHEG-5 elementary action
    public void toggleItem(  
        short itemIndex)  
    throws MhegException;

    // Correspond to the ScrollItems MHEG-5 elementary action
    public void scrollItems(  
        short itemsToScroll)  
    throws MhegException;

    // Correspond to the SetFirstItem MHEG-5 elementary action
    public void setFirstItem(  
        short itemIndex)  
    throws MhegException;

    // Correspond to the getFirstItem MHEG-5 elementary action
    public Integer getFirstItem()  
    throws MhegException;

    // Correspond to the getListSize MHEG-5 elementary action
    public Integer getListSize()  
    throws MhegException;

    // -------------------------------
    abstract public class Visible extends Presentable
/* Constructors */
    protected Visible();
/* Methods */
// Correspond to the SetPosition MHEG-5 elementary action
    public void setPosition(
        short xPosition,
        short yPosition)
    throws MhegException;
// Correspond to the GetPosition MHEG-5 elementary action
    public void getPosition(
        Integer xPosition,
        Integer yPosition)
    throws MhegException;
// Correspond to the SetBoxSize MHEG-5 elementary action
// The xBoxSize and yBoxSize parameter values shall be positive and different from zero
    public void setBoxSize(
        short xBoxSize,
        short yBoxSize)
    throws MhegException;
// Correspond to the GetBoxSize MHEG-5 elementary action
    public void getBoxSize(
        Integer xBoxSize,
        Integer yBoxSize)
    throws MhegException;
// Correspond to the BringToFront MHEG-5 elementary action
    public void bringToFront()
    throws MhegException;
// Correspond to the SendToBack MHEG-5 elementary action
    public void sendToBack()
    throws MhegException;
// Correspond to the PutBefore MHEG-5 elementary action
    public void putBefore(
        ObjectReference visibleReference)
    throws MhegException;
// Correspond to the PutBehind MHEG-5 elementary action
    public void putBehind(
        ObjectReference visibleReference)
    throws MhegException;
// Correspond to the SetPaletteRef MHEG-5 elementary action
    public void setPaletteRef(
        ObjectReference paletteReference)
    throws MhegException;
// Retrieve the value of the PaletteRef attribute
    public ObjectReference getPaletteRef()
    throws MhegException;
}

public class Bitmap extends Visible
{
/* Constructors */
    protected Bitmap();
/* Methods */
// Correspond to the ScaleBitmap MHEG-5 elementary action
// The xScale and yScale parameter values shall be positive and different from zero
    public void scaleBitmap(
        short xScale,
        short yScale)
    throws MhegException;
// Correspond to the SetTransparency MHEG-5 elementary action
// The transparency parameter value shall be within the range [0,100]
    public void setTransparency(
        byte transparency)
    throws MhegException;
// Retrieve the value of the Transparency attribute
public Integer getTransparency()
    throws MhegException;
}

// ----------------------------------------------------------------------------------------------
public class LineArt extends Visible
{
    /* Constructors */
    protected LineArt();
    /* Methods */
    // Correspond to the SetLineWidth MHEG-5 elementary action
    // The lineWidth parameter value shall be positive and different from zero
    public void setLineWidth(
        byte lineWidth)
        throws MhegException;
    // Correspond to the SetLineStyle MHEG-5 elementary action
    // The lineStyle parameter value shall be 1 for solid, 2 for dashed, 3 for dotted
    public void setLineStyle(
        byte lineStyle)
        throws MhegException;
    // Correspond to the SetLineColour MHEG-5 elementary action
    // When neither a colour index nor an absolute colour is specified, the fill-in colour shall be set to
    // transparent
    public void setFillColour()
        throws MhegException;
    public void setFillColour(
        byte colourIndex)
        throws MhegException;
    public void setFillColour(
        byte[] absoluteColour)
        throws MhegException;
}

// ----------------------------------------------------------------------------------------------
public class Rectangle extends LineArt
{
    /* Constructors */
    protected Rectangle();
}

// ----------------------------------------------------------------------------------------------
public class DynamicLineArt extends LineArt
{
    /* Constructors */
    protected DynamicLineArt();
    /* Methods */
    // Correspond to the GetLineWidth MHEG-5 elementary action
    public Integer getLineWidth()
        throws MhegException;
    // Correspond to the GetLineStyle MHEG-5 elementary action
    public Integer getLineStyle()
        throws MhegException;
    // Correspond to the GetLineColour MHEG-5 elementary action
    // The returned value shall be of one of the following types: Integer (in which case it represents an index) or
    // byte[] (in which case it represents the absolute colour)
public Object getLineColour()
    throws MhegException;

// Correspond to the GetFillColour MHEG-5 elementary action
// The returned value shall be of one of the following types: Integer (in which case it
// represents an index) or
// byte[] (in which case it represents the absolute colour); a returned null object reference
// indicates no fill-in
// colour (transparent)
public Object getFillColour()
    throws MhegException;

public void drawArc(
    short x,
    short y,
    short ellipseWidth,
    short ellipseHeight,
    short startAngle,
    short arcAngle)
    throws MhegException;

// Correspond to the DrawArc MHEG-5 elementary action
public void drawSector(
    short x,
    short y,
    short ellipseWidth,
    short ellipseHeight,
    short startAngle,
    short arcAngle)
    throws MhegException;

// Correspond to the DrawSector MHEG-5 elementary action
public void drawLine(
    short x1,
    short y1,
    short x2,
    short y2)
    throws MhegException;

// Correspond to the DrawLine MHEG-5 elementary action
public void drawOval(
    short x,
    short y,
    short ellipseWidth,
    short ellipseHeight)
    throws MhegException;

// Correspond to the DrawOval MHEG-5 elementary action
public void drawPolygon(
    short[][] pointList)
    throws MhegException;

// Correspond to the DrawPolygon MHEG-5 elementary action
public void drawPolyline(
    short[][] pointList)
    throws MhegException;

// Correspond to the DrawPolyline MHEG-5 elementary action
public void drawRectangle(
    short x1,
    short y1,
    short x2,
    short y2)
    throws MhegException;

// Correspond to the DrawRectangle MHEG-5 elementary action
public void clear()
    throws MhegException;

// Correspond to the Clear MHEG-5 elementary action
public class Text extends Visible
{

    /* Constructors */
    protected Text();

    /* Methods */

    public Object getLineColour()
    throws MhegException;

// Correspond to the GetLineColour MHEG-5 elementary action
// The returned value shall be of one of the following types: Integer (in which case it
// represents an index) or
// byte[] (in which case it represents the absolute colour); a returned null object reference
// indicates no fill-in
// colour (transparent)
public Object getFillColour()
    throws MhegException;

// Correspond to the GetFillColour MHEG-5 elementary action
// The returned value shall be of one of the following types: Integer (in which case it
// represents an index) or
// byte[] (in which case it represents the absolute colour); a returned null object reference
// indicates no fill-in
// colour (transparent)
public Object getFillColour()
    throws MhegException;
public Object getTextContent()
    throws MhegException;

public byte[] getTextData()
    throws MhegException;

public ObjectReference getFontRef()
    throws MhegException;

protected Stream();

/* Methods */

public void setCounterTrigger(
    short triggerIdentifier)
    throws MhegException;

public void setCounterTrigger(
    short triggerIdentifier,
    byte counterValue)
    throws MhegException;

public Integer getCounterTrigger(
    short triggerIdentifier)
    throws MhegException;

public void setSpeed(
    byte nominator)
    throws MhegException;

public void setSpeed(
    byte nominator,
    byte denominator)
    throws MhegException;

public Integer getSpeed(
    Integer nominator,
    Integer denominator)
    throws MhegException;

public void setCounterPosition(
    short counterPosition)
    throws MhegException;

public Integer getCounterPosition()
    throws MhegException;

public void setCounterEndPosition(
    short counterEndPosition)
    throws MhegException;

// Correspond to the GetTextData MHEG-5 elementary action
// Correspond to the SetFontRef MHEG-5 elementary action
// Correspond to the SetCounterTrigger MHEG-5 elementary action
// Correspond to the SetSpeed MHEG-5 elementary action
// Correspond to the SetCounterPosition MHEG-5 elementary action
// Correspond to the SetCounterEndPosition MHEG-5 elementary action
// Retrieve the value of the CounterEndPosition attribute
   public Integer getCounterEndPosition()
       throws MhegException;
   }

   -------------------------------------------------------------------------------
   public class Audio extends Presentable
   {
       /* Constructors */
       protected Audio();
       /* Methods */
       // Correspond to the SetVolume MHEG-5 elementary action
       public void setVolume(
           byte volume)
           throws MhegException;
       // Correspond to the GetVolume MHEG-5 elementary action
       public Integer getVolume()
           throws MhegException;
   }
   -------------------------------------------------------------------------------
   public class Video extends Visible
   {
       /* Constructors */
       protected Video();
       /* Methods */
       // Correspond to the ScaleVideo MHEG-5 elementary action
       // The xScale and yScale parameter values shall be positive and different from zero
       public void scaleVideo(
           short xScale,
           short yScale)
           throws MhegException;
   }
   -------------------------------------------------------------------------------
   public class RTGraphics extends Visible
   {
       /* Constructors */
       protected RTGraphics();
   }
   // -------------------------------------------------------------------------------
   public interface Interactible
   {
       /* Methods */
       // Correspond to the SetInteractionStatus MHEG-5 elementary action
       public void setInteractionStatus(
           boolean interactionStatus)
           throws MhegException;
       // Correspond to the GetInteractionStatus MHEG-5 elementary action
       public Boolean getInteractionStatus()
           throws MhegException;
       // Correspond to the SetHighlightStatus MHEG-5 elementary action
       public void setHighlightStatus(
           boolean highlightStatus)
           throws MhegException;
       // Correspond to the GetHighlightStatus MHEG-5 elementary action
       public Boolean getHighlightStatus()
           throws MhegException;
   }
   // -------------------------------------------------------------------------------
   public class Slider extends Visible implements Interactible
   {
protected Slider();

/* Methods */

// Correspond to the implementation of the Interactible.setInteractionStatus method
public void setInteractionStatus(
    boolean interactionStatus)
    throws MhegException;

// Correspond to the implementation of the Interactible.getInteractionStatus method
public boolean getInteractionStatus()
    throws MhegException;

// Correspond to the implementation of the Interactible.setHighlightStatus method
public void setHighlightStatus(
    boolean highlightStatus)
    throws MhegException;

// Correspond to the implementation of the Interactible.getHighlightStatus method
public boolean getHighlightStatus()
    throws MhegException;

// Correspond to the Step MHEG-5 elementary action
public void step(
    byte nbOfSteps)
    throws MhegException;

// Correspond to the SetSliderValue MHEG-5 elementary action
public void setSliderValue(
    byte sliderValue)
    throws MhegException;

// Correspond to the GetSliderValue MHEG-5 elementary action
public byte getSliderValue()
    throws MhegException;

// Correspond to the SetPortion MHEG-5 elementary action
public void setPortion(
    byte portion)
    throws MhegException;

// Correspond to the GetPortion MHEG-5 elementary action
public byte getPortion()
    throws MhegException;

// Correspond to the SetOverwriteMode MHEG-5 elementary action
public void setOverwriteMode(
    boolean overwriteMode)
    throws MhegException;

// Correspond to the GetOverwriteMode MHEG-5 elementary action
public boolean getOverwriteMode()
public Boolean getOverwriteMode()
    throws MhegException;

// Correspond to the SetEntryPoint MHEG-5 elementary action
// The entryPoint parameter value shall be positive or equal to zero
public void setEntryPoint(
    byte entryPoint)
    throws MhegException;

// Correspond to the GetEntryPoint MHEG-5 elementary action
public Integer getEntryPoint()
    throws MhegException;

}  

public class HyperText extends Text implements Interactible
{

/* Constructors */
    protected HyperText();

/* Methods */

// Correspond to the implementation of the Interactible.setInteractionStatus method
public void setInteractionStatus(
    boolean interactionStatus)
    throws MhegException;

// Correspond to the implementation of the Interactible.getInteractionStatus method
public Boolean getInteractionStatus()
    throws MhegException;

// Correspond to the implementation of the Interactible.setHighlightStatus method
public void setHighlightStatus(
    boolean highlightStatus)
    throws MhegException;

// Correspond to the implementation of the Interactible.getHighlightStatus method
public Boolean getHighlightStatus()
    throws MhegException;

// Correspond to the GetLastAnchorFired MHEG-5 elementary action
public byte[] getLastAnchorFired()
    throws MhegException;

}  

abstract public class Button extends Visible implements Interactible
{

/* Constructors */
    protected Button();

/* Methods */

// Correspond to the implementation of the Interactible.setInteractionStatus method
public void setInteractionStatus(
    boolean interactionStatus)
    throws MhegException;

// Correspond to the implementation of the Interactible.getInteractionStatus method
public Boolean getInteractionStatus()
    throws MhegException;

// Correspond to the implementation of the Interactible.setHighlightStatus method
public void setHighlightStatus(
    boolean highlightStatus)
    throws MhegException;

// Correspond to the implementation of the Interactible.getHighlightStatus method
public Boolean getHighlightStatus()
    throws MhegException;

// Correspond to the Select MHEG-5 elementary action
public void select()
    throws MhegException;

// Correspond to the Deselect MHEG-5 elementary action
public void deselect()
    throws MhegException;
public class Hotspot extends Button {
    /* Constructors */
    protected Hotspot();
}

public class PushButton extends Button {
    /* Constructors */
    protected PushButton();
    /* Methods */
    // Correspond to the SetLabel MHEG-5 elementary action
    public void setLabel(byte[] label) throws MhegException;
    // Correspond to the GetLabel MHEG-5 elementary action
    public byte[] getLabel() throws MhegException;
}

public class SwitchButton extends PushButton {
    /* Constructors */
    protected SwitchButton();
    /* Methods */
    // Correspond to the GetSelectionStatus MHEG-5 elementary action
    public Boolean getSelectionStatus() throws MhegException;
    // Correspond to the Toggle MHEG-5 elementary action
    public void toggle() throws MhegException;
}
6 Map between MHEG-5 elementary actions and MHEG-5 API operations

The following table describes how the MHEG-5 API maps MHEG-5 classes to Java classes and interfaces, and MHEG-5 elementary actions to Java methods. Table 1 is not normative.

Table 1: Map between MHEG-5 elementary actions and MHEG-5 API operations

<table>
<thead>
<tr>
<th>MHEG-5 class</th>
<th>MHEG-5 elementary action</th>
<th>API operation (overloads)</th>
<th>API interface or class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>Root</td>
<td>Root (abstract class)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetAvailabilityStatus</td>
<td>getAvailabilityStatus</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetRunningStatus</td>
<td>getRunningStatus</td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>SetCachePriority</td>
<td>setCachePriority</td>
<td>Group (abstract class)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>getCachePriority</td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>StorePersistent</td>
<td>Application (class)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ReadPersistent</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Launch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spawn</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quit</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LockScreen</td>
<td>lockScreen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UnlockScreen</td>
<td>unlockScreen</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>getLockCount</td>
<td></td>
</tr>
<tr>
<td>OpenConnection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CloseConnection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetEngineSupport</td>
<td>getEngineSupport</td>
<td></td>
</tr>
<tr>
<td>Scene</td>
<td>TransitionTo</td>
<td>transitionTo (4)</td>
<td>Scene (class)</td>
</tr>
<tr>
<td></td>
<td>SetTimer</td>
<td>setTimer (3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>getTimer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SendEvent</td>
<td>sendEvent (4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetCursorShape</td>
<td>setCursorShape (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>getCursorShape</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetCursorPosition</td>
<td>setCursorPosition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetCursorPosition</td>
<td>getCursorPosition</td>
<td></td>
</tr>
<tr>
<td>Ingredient</td>
<td>SetData</td>
<td>setData (5)</td>
<td>Ingredient (abstract class)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>getData</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Clone</td>
<td>clone</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Preload</td>
<td>preload</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unload</td>
<td>unload</td>
<td></td>
</tr>
<tr>
<td>Link</td>
<td>Activate</td>
<td>activate</td>
<td>Link (class)</td>
</tr>
<tr>
<td></td>
<td>Deactivate</td>
<td>deactivate</td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>Call</td>
<td>call</td>
<td>Program (abstract class)</td>
</tr>
<tr>
<td></td>
<td>Fork</td>
<td>fork</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stop</td>
<td>stop</td>
<td></td>
</tr>
<tr>
<td>ResidentProgram</td>
<td></td>
<td></td>
<td>ResidentProgram (class)</td>
</tr>
<tr>
<td>RemoteProgram</td>
<td></td>
<td></td>
<td>RemoteProgram (class)</td>
</tr>
<tr>
<td>InterchangedProgram</td>
<td></td>
<td></td>
<td>InterchangedProgram (class)</td>
</tr>
<tr>
<td>Palette</td>
<td></td>
<td></td>
<td>Palette (class)</td>
</tr>
<tr>
<td>Font</td>
<td></td>
<td></td>
<td>Font (class)</td>
</tr>
<tr>
<td>CursorShape</td>
<td></td>
<td></td>
<td>CursorShape (class)</td>
</tr>
<tr>
<td>Variable</td>
<td>SetVariable</td>
<td></td>
<td>Variable (abstract class)</td>
</tr>
<tr>
<td></td>
<td>TestVariable</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(continued)
<table>
<thead>
<tr>
<th>MHEG-5 class</th>
<th>MHEG-5 elementary action</th>
<th>API operation (overloads)</th>
<th>API interface or class</th>
</tr>
</thead>
<tbody>
<tr>
<td>BooleanVariable</td>
<td>setVariable</td>
<td>getVariable</td>
<td>BooleanVariable (class)</td>
</tr>
<tr>
<td>IntegerVariable</td>
<td>setVariable</td>
<td>getVariable</td>
<td>IntegerVariable (class)</td>
</tr>
<tr>
<td>Add</td>
<td>add</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subtract</td>
<td>subtract</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiply</td>
<td>multiply</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Divide</td>
<td>divide</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modulo</td>
<td>modulo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OctetStringVariable</td>
<td>setVariable</td>
<td>getVariable</td>
<td>OctetStringVariable (class)</td>
</tr>
<tr>
<td>Append</td>
<td>append</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ObjectRefVariable</td>
<td>setVariable</td>
<td>getVariable</td>
<td>ObjectRefVariable (class)</td>
</tr>
<tr>
<td>ContentRefVariable</td>
<td>setVariable</td>
<td>getVariable</td>
<td>ContentRefVariable (class)</td>
</tr>
<tr>
<td>Presentable</td>
<td>Run</td>
<td>run</td>
<td>Presentable (abstract class)</td>
</tr>
<tr>
<td>TokenManager</td>
<td>Move</td>
<td>move</td>
<td>TokenManager (interface)</td>
</tr>
<tr>
<td></td>
<td>moveTo</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>getTokenPosition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TokenGroup</td>
<td>CallActionSlot</td>
<td>callActionSlot</td>
<td>TokenGroup (class)</td>
</tr>
<tr>
<td></td>
<td>AddItem</td>
<td>addItem</td>
<td>TemplateGroup (class)</td>
</tr>
<tr>
<td></td>
<td>dellItem</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>getListItem</td>
<td>getListItem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getCellItem</td>
<td>getCellItem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getItemStatus</td>
<td>getItemStatus</td>
<td></td>
</tr>
<tr>
<td></td>
<td>selectItem</td>
<td>selectItem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>deselectItem</td>
<td>deselectItem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>toggleItem</td>
<td>toggleItem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>scrollItems</td>
<td>scrollItems</td>
<td></td>
</tr>
<tr>
<td></td>
<td>setFirstItem</td>
<td>setFirstItem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getFirstItem</td>
<td>getFirstItem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getListSize</td>
<td>getListSize</td>
<td></td>
</tr>
<tr>
<td>Visible</td>
<td>setPosition</td>
<td>setPosition</td>
<td>Visible (abstract class)</td>
</tr>
<tr>
<td></td>
<td>getPosition</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>setBoxSize</td>
<td>setBoxSize</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getBoxSize</td>
<td>getBoxSize</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bringToFront</td>
<td>bringToFront</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sendToBack</td>
<td>sendToBack</td>
<td></td>
</tr>
<tr>
<td></td>
<td>putBefore</td>
<td>putBefore</td>
<td></td>
</tr>
<tr>
<td></td>
<td>putBehind</td>
<td>putBehind</td>
<td></td>
</tr>
<tr>
<td></td>
<td>setPaletteRef</td>
<td>setPaletteRef</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getPaletteRef</td>
<td>getPaletteRef</td>
<td></td>
</tr>
<tr>
<td>Bitmap</td>
<td>ScaleBitmap</td>
<td>scaleBitmap</td>
<td>Bitmap (class)</td>
</tr>
<tr>
<td></td>
<td>setTransparency</td>
<td>setTransparency</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getTransparency</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1 (concluded): Map between MHEG-5 elementary actions and MHEG-5 API operations

<table>
<thead>
<tr>
<th>MHEG-5 class</th>
<th>MHEG-5 elementary action</th>
<th>API operation (overloads)</th>
<th>API interface or class</th>
</tr>
</thead>
<tbody>
<tr>
<td>LineArt</td>
<td>SetLineWidth</td>
<td>setLineWidth</td>
<td>LineArt (class)</td>
</tr>
<tr>
<td></td>
<td>SetLineStyle</td>
<td>setLineStyle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetLineColour</td>
<td>setLineColour (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetFillColour</td>
<td>setFillColour (3)</td>
<td></td>
</tr>
<tr>
<td>Rectangle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DynamicLineArt</td>
<td>GetLineWidth</td>
<td>getLineWidth</td>
<td>DynamicLineArt (class)</td>
</tr>
<tr>
<td></td>
<td>GetLineStyle</td>
<td>getLineStyle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetLineColour</td>
<td>getLineColour</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetFillColour</td>
<td>getFillColour</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DrawArc</td>
<td>drawArc</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DrawSector</td>
<td>drawSector</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DrawLine</td>
<td>drawLine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DrawOval</td>
<td>drawOval</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DrawPolygon</td>
<td>drawPolygon</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DrawPolyline</td>
<td>drawPolyline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DrawRectangle</td>
<td>drawRectangle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Clear</td>
<td>clear</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>GetTextContent</td>
<td>getTextContent</td>
<td>Text (class)</td>
</tr>
<tr>
<td></td>
<td>GetTextData</td>
<td>getTextData</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetFontRef</td>
<td>setFontRef (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getFontRef</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stream</td>
<td>SetCounterTrigger</td>
<td>setCounterTrigger (2)</td>
<td>Stream (class)</td>
</tr>
<tr>
<td></td>
<td>GetCounterTrigger</td>
<td>getCounterTrigger</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetSpeed</td>
<td>setSpeed (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getSpeed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetCounterPosition</td>
<td>setCounterPosition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getCounterPosition</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetCounterEndPosition</td>
<td>setCounterEndPosition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>getCounterEndPosition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td>SetVolume</td>
<td>setVolume</td>
<td>Audio (class)</td>
</tr>
<tr>
<td></td>
<td>GetVolume</td>
<td>getVolume</td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td>ScaleVideo</td>
<td>scaleVideo</td>
<td>Video (class)</td>
</tr>
<tr>
<td>RTGraphics</td>
<td></td>
<td></td>
<td>RTGraphics (class)</td>
</tr>
<tr>
<td>Interactible</td>
<td>SetInteractionStatus</td>
<td>setInteractionStatus</td>
<td>Interactible (interface)</td>
</tr>
<tr>
<td></td>
<td>GetInteractionStatus</td>
<td>getInteractionStatus</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetHighlightStatus</td>
<td>setHighlightStatus</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetHighlightStatus</td>
<td>getHighlightStatus</td>
<td></td>
</tr>
<tr>
<td>Slider</td>
<td>Step</td>
<td>step</td>
<td>Slider (class)</td>
</tr>
<tr>
<td></td>
<td>SetSliderValue</td>
<td>setSliderValue</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetSliderValue</td>
<td>getSliderValue</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetPortion</td>
<td>setPortion</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetPortion</td>
<td>getPortion</td>
<td></td>
</tr>
<tr>
<td>EntryField</td>
<td>SetOverwriteMode</td>
<td>setOverwriteMode</td>
<td>EntryField (class)</td>
</tr>
<tr>
<td></td>
<td>GetOverwriteMode</td>
<td>getOverwriteMode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SetEntryPoint</td>
<td>setEntryPoint</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetEntryPoint</td>
<td>getEntryPoint</td>
<td></td>
</tr>
<tr>
<td>HyperText</td>
<td>GetLastAnchorFired</td>
<td>getLastAnchorFired</td>
<td>HyperText (class)</td>
</tr>
<tr>
<td>Button</td>
<td>Select</td>
<td>select</td>
<td>Button (abstract class)</td>
</tr>
<tr>
<td></td>
<td>Deselect</td>
<td>deselect</td>
<td></td>
</tr>
<tr>
<td>Hotspot</td>
<td></td>
<td></td>
<td>Hotspot (class)</td>
</tr>
<tr>
<td>PushButton</td>
<td>SetLabel</td>
<td>setLabel</td>
<td>PushButton (class)</td>
</tr>
<tr>
<td></td>
<td>GetLabel</td>
<td>getLabel</td>
<td></td>
</tr>
<tr>
<td>SwitchButton</td>
<td>GetSelectionStatus</td>
<td>getSelectionStatus</td>
<td>SwitchButton (class)</td>
</tr>
<tr>
<td></td>
<td>Toggle</td>
<td>toggle</td>
<td></td>
</tr>
</tbody>
</table>
## History

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Document</th>
<th>Date Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 1997</td>
<td>Vote</td>
<td>V 9735:</td>
<td>1997-06-17 to 1997-08-29</td>
</tr>
<tr>
<td>September 1997</td>
<td>First Edition</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>