SLOVENSKI STANDARD
SIST ETS 300 777-3 E1:2003

01-december-2003

Terminalska oprema (TE) – Protokoli sistema konec-konec za multimedijske informacijske poiskavne storitve – 3. del: Aplikacijski programski vmesnik (API) za MHEG-5

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

Ta slovenski standard je istoveten z: **ETS 300 777-3 Edition 1**

ICS:

33.160.60	Več predstavni (multimedijski) sistemi in oprema za telekonference	Multimedia systems and teleconferencing equipment
35.180	Terminalska in druga periferna oprema IT	IT Terminal and other peripheral equipment

SIST ETS 300 777-3 E1:2003 **en**

EUROPEAN TELECOMMUNICATION STANDARD

ETS 300 777-3

September 1997

Source: MTA

Reference: DE/MTA-011057-3
Formerly: DE/TE-010157-3

ICS: 33.020

Key words: API, MHEG, multimedia, terminal

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

Contents

Blank page

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".

Transposition dates	
Date of adoption:	5 September 1997
Date of latest announcement of this ETS (doa):	31 December 1997
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	30 June 1998
Date of withdrawal of any conflicting National Standard (dow):	30 June 1998

Blank page

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.

[1] ISO/IEC IS 13522-5 (1997): "Information technology - Coding of Multimedia and Hypermedia Information - Part 5: Support for Base-Level Interactive Applications".

[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".

[4] ISO/IEC 13522-6 : "Information technology - Coding of Multimedia and Hypermedia Information - Part 6: Support for enhanced interactive 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:

- ISO/IEC IS 13522-5 [1] MHEG-5;
- 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

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

// ------------------------------------------------------------------------
// Useful definitions
// ------------------------------------------------------------------------
public class ObjectReference
{

/* Attributes */

// The groupIdentifier attribute is optional (it may be empty)
    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;

}
```