



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.175

(02/98)

SERIES T: TERMINALS FOR TELEMATIC SERVICES

Application Programming Interface (API) for MHEG-5

ITU-T Recommendation T.175

(Previously CCITT Recommendation)

ITU-T T-SERIES RECOMMENDATIONS
TERMINALS FOR TELEMATIC SERVICES



For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION T.175**APPLICATION PROGRAMMING INTERFACE (API) FOR MHEG-5****Summary**

This Recommendation specifies the Application Programming Interface (API) for the manipulation of multimedia and hypermedia information objects.

MHEG part 5 (Recommendation T.172) 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.

Source

ITU-T Recommendation T.175 was prepared by ITU-T Study Group 16 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 6th of February 1998.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 1998

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

	Page
1 Scope	1
2 Normative references.....	1
3 Definitions and abbreviations.....	1
3.1 Definitions	1
3.2 Abbreviations	2
4 Overview	2
4.1 The DAVIC application interchange format	2
4.2 Core set of Java APIs	3
5 The MHEG-5 API	4
6 Map between MHEG-5 elementary actions and MHEG-5 API operations	22

Recommendation T.175**APPLICATION PROGRAMMING INTERFACE (API) FOR MHEG-5***(Geneva, 1998)***1 Scope**

This Recommendation specifies the Application Programming 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 (Recommendation T.172) 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

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- [1] ISO/IEC 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 DIS 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 Recommendation the definition of the standards referenced below apply. Should any ambiguity occur, definitions of the following standards apply, in decreasing order:

- ISO/IEC 13522-5 [1] MHEG-5;
- any other standard part of ISO/IEC 13522 MHEG.

3.1.1 Application Programming 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.

**Documents**

Document Sharing Hub

Recommendation T.175 (02/98)

1

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

3.1.3 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 Recommendation.

3.1.4 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

This Recommendation uses the following abbreviations:

API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
DAVIC	Digital Audio Visual Council
DSM-CC	Digital Storage Media Command and Control
EBNF	Extended Backus-Naur Form
JTC	Joint Technical Committee
MHEG	Multimedia and Hypermedia information coding Experts Group
SI	Service Information
STU	Set Top Unit
VM	Virtual Machine

4 Overview

The following subclause positions the API defined by this Recommendation 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 InterchangedProgram class to encapsulate Java¹ VM code, according to the semantics and encoding defined by ISO/IEC DIS 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*.

¹ Java is a trademark or a registered trademark of Sun Microsystems, Inc.

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 standardised 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, mathematic 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 Recommendation, 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 groupIdIdentifier attribute is optional (it may be empty)
    public byte[] groupIdIdentifier;
    public int objectNumber;

    /* Constructors */

    public ObjectReference();
    public ObjectReference(
        int objectNumber);
    public ObjectReference(
        byte[] groupIdIdentifier,
        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;
}

```