

ETSI TS 102 819 V1.1.1 (2003-01)

Technical Specification

Digital Video Broadcasting (DVB); Globally Executable MHP (GEM) Specification 1.0.0

European Broadcasting Union



Union Européenne de Radio-Télévision



Reference

DTS/JTC-DVB-145

Keywords

API, broadcasting, digital, DVB, interaction,
multimedia, profile, video

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

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

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, send your comment to:

editor@etsi.org

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 2003.

© European Broadcasting Union 2003.

All rights reserved.

DECT™, **PLUGTESTS™** and **UMTS™** are Trade Marks of ETSI registered for the benefit of its Members.
TIPHON™ and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Contents

Intellectual Property Rights	8
Foreword.....	8
1 Scope	9
2 References	9
3 Definitions and abbreviations.....	10
3.1 Definitions	10
3.1.1 Definitions from MHP	10
3.1.2 Definitions introduced by GEM.....	10
3.2 Abbreviations	10
4 General considerations and conventions	11
4.1 General considerations	11
4.1.1 Purpose	11
4.1.2 Format.....	11
4.1.3 Inclusion of MHP features.....	12
4.1.3.1 Subsetting prohibited	12
4.1.3.2 Supersetting permitted	12
4.1.4 Addition of non-GEM interfaces	12
4.1.4.1 DVB-J enumerations.....	12
4.1.5 Application areas	12
4.1.6 Profiles.....	12
4.1.7 Full compliance with the present document	12
4.2 Conventions.....	13
4.2.1 References within the MHP specification.....	13
4.2.2 Terminology in the MHP specification.....	13
4.2.2.1 MHP	13
4.2.2.2 Resident navigator.....	13
4.2.2.3 DVB service	13
4.2.3 Inclusion of clauses of the MHP specification.....	13
5 Basic architecture	13
6 Transport protocols.....	14
6.1 Introduction	14
6.2 Broadcast channel protocols.....	14
6.2.1 MPEG-2 transport stream	14
6.2.2 MPEG-2 clauses	14
6.2.3 DSM-CC private data	14
6.2.4 DSM-CC data carousel	14
6.2.5 DSM-CC user-to-user object carousel	14
6.2.5.1 DVB-J class files.....	14
6.2.5.2 DVB-HTML document files	15
6.2.5.3 Loss of carousel behaviour.....	15
6.2.6 Protocol for delivery of IP multicast over the broadcast channel	15
6.2.7 Internet Protocol (IP)	15
6.2.8 User Datagram Protocol (UDP).....	15
6.2.9 DVB service information	15
6.3 Interaction channel protocols	15
7 Content formats	15
7.1 Static formats.....	15
7.2 Broadcast streaming formats	16
7.2.1 Audio	16
7.2.2 Video	16
7.2.3 Subtitles	16
7.3 Resident fonts.....	16

7.4	Downloadable fonts.....	16
7.5	Colour representation	16
7.6	MIME types.....	16
8	DVB-HTML.....	16
9	Application model.....	17
9.1	Broadcast GEM applications.....	17
9.2	DVB-J model.....	17
9.3	DVB-HTML model.....	17
9.4	Inter-application resource management.....	17
10	Application signalling	17
10.1	Introduction	17
10.1.1	Summary of requirements on common signalling	18
10.1.2	Summary of additional signalling for DVB-J applications.....	18
10.2	Program specific information.....	18
10.3	Locators within an application description.....	18
10.4	Application description	18
10.4.1	Application description transmission and monitoring	18
10.4.2	Visibility of application description.....	19
10.4.3	Content of the application description	19
10.4.4	Applications from previously selected services.....	20
10.5	DVB-J specific application description.....	20
10.5.1	General.....	20
10.5.2	Content of DVB-J application description.....	20
11	DVB-J platform.....	21
11.1	The virtual machine.....	21
11.2	General issues.....	21
11.3	Fundamental DVB-J APIs.....	21
11.4	Presentation APIs	22
11.5	Data access APIs	22
11.6	Service information and selection APIs	22
11.6.1	DVB service information API	22
11.6.2	Service selection API.....	23
11.6.3	Tuning API	23
11.6.4	Conditional access API.....	23
11.6.5	Protocol independent SI API	23
11.7	Common infrastructure APIs.....	23
11.7.1	APIs to support DVB-J application lifecycle.....	23
11.7.2	Application discovery and launching APIs.....	23
11.7.3	Inter-application communication API.....	24
11.7.4	Basic MPEG concepts	24
11.7.5	Resource notification.....	24
11.7.6	Content referencing.....	24
11.7.7	Common error reporting	25
11.8	Security	25
11.9	Other APIs.....	25
11.9.1	Timer support.....	25
11.9.2	User settings and preferences API.....	25
11.9.3	Profile and version properties	25
11.10	Java permissions.....	26
11.11	Content referencing	26
11.11.0	General.....	26
11.11.1	Transport stream	26
11.11.2	Network	26
11.11.3	Bouquet.....	26
11.11.4	Service	26
11.11.4.1	MPEG/GEM specific service	26
11.11.4.2	Generic service.....	27
11.11.5	Program event.....	27
11.11.6	MPEG elementary stream.....	27

11.11.7	File	28
11.11.8	Directory	28
11.11.9	Drip feed decoder	28
11.11.10	Irrelevant.....	28
11.11.11	Methods working on many locator types	28
11.11.12	Support for the HTTP Protocol in DVB-J	28
12	Security.....	29
12.1	Introduction	29
12.1.1	Overview of the security framework for applications.....	29
12.1.2	Overview of return channel security	29
12.1.3	Extensions to MHP application signing framework	29
12.2	Authentication of applications.....	29
12.3	Message transport.....	29
12.4	Detail of application authentication messages.....	30
12.5	Profile of X.509 certificates for authentication of applications.....	30
12.6	Security policy for applications.....	30
12.7	Example of creating an application that can be authenticated.....	30
12.8	GEM/MHP certification procedures.....	30
12.9	Certificate management.....	31
12.10	Security on the return channel.....	31
12.11	The internet profile of X.509 (informative).....	31
12.12	Platform minima.....	31
13	Graphics reference model.....	31
13.0	General	31
13.1	Supported graphics resolutions.....	31
13.2	Broadcast streaming formats	31
13.3	Subtitles.....	31
14	System integration aspects	32
14.1	Namespace mapping.....	32
14.2	Reserved names.....	32
14.3	XML notation	32
14.4	Network signalling	32
14.5	Text encoding of application identifiers.....	32
14.6	Reserved names for persistent storage.....	32
14.7	Files and file names.....	32
14.8	Locators and content referencing	32
14.9	Service identification.....	33
14.10	CA system	33
15	Detailed platform profile definitions	33
15.0	General	33
15.1	PNG - restrictions.....	35
15.2	Minimum media formats supported by DVB-J APIs	35
15.3	JPEG - restrictions.....	35
15.4	Locale support	35
15.5	Video raster format dependencies	35
15.5.1	Standard Definition (PAL/SECAM or NTSC resolution).....	35
15.5.1.1	Logical pixel resolution.....	35
16	Registry of constants	36
16.1	System constants	36
16.2	DVB-J constants.....	36
Annex A (normative):	External references; errata, clarifications and exemptions	37
Annex B (normative):	Broadcast filesystem and trigger transport.....	38
B.0	General	38
B.1	Service domain.....	38
B.2	Filesystem requirements.....	38

B.2.1	Static requirements	38
B.2.1.1	Caching behaviour	39
B.2.2	Filesystem updates	39
B.3	Stream description	39
B.4	Trigger signalling	40
B.4.0	General	40
B.4.1	Trigger object	40
B.4.2	Trigger event	40
B.4.2.1	Extrapolation of NPT values	41
B.4.2.2	Monitoring of trigger events	41
Annex C (informative): References		42
C.1	Informative references from MHP	42
C.2	Additional informative references	42
Annex D (normative): Text presentation		43
Annex E (normative): Character set		44
Annex F (informative): Authoring and implementation guidelines		45
Annex G (normative): Minimum platform capabilities		46
G.1	Graphics	46
G.2	Audio	46
G.3	Video	46
G.4	Resident fonts and text rendering	46
G.5	Input events	46
G.6	Memory	46
G.7	Other resources	47
Annex H (normative): Extensions		48
Annex I (normative): DVB-J fundamental classes		49
Annex J (normative): DVB-J event API		50
Annex K (normative): DVB-J persistent storage API		51
Annex L (normative): User settings and preferences API		52
Annex M: Void		53
Annex N (normative): Streamed media API extensions		54
Annex O (normative): Integration of the JavaTV SI API		55
Annex P (normative): Broadcast transport protocol access		56
P.1	Overview	56
P.2	The org.dvb.dsmcc package	57
P.2.0	General	57
P.2.1	DSMCCObject	57
P.2.2	DSMCCStream	57
P.2.2.1	isAudio() method	57
P.2.2.2	isData() method	57
P.2.2.3	isMPEGProgram() method	57

P.2.2.4	isVideo() method	57
P.2.3	DSMCCStreamEvent	57
P.2.4	InvalidFormatException	57
P.2.5	ServiceDomain	58
P.2.5.1	ServiceDomain.attach(byte[])	58
P.2.5.2	ServiceDomain.attach(Locator) and attach(Locator, int).....	58
P.2.5.3	ServiceDomain.getLocator().....	58
P.2.5.4	ServiceDomain.getNSAPAddress().....	58
P.2.5.5	ServiceDomain.getURL(Locator).....	58
P.2.6	ServiceXFRErrorEvent	59
P.2.7	ServiceXFRException	59
P.2.8	ServiceXFRReference	59
P.2.9	StreamEvent	59
Annex Q (normative):	Datagram socket buffer control.....	60
Annex R (normative):	DVB-J return channel connection management API	61
Annex S (normative):	Application listing and launching.....	62
Annex T (normative):	Permissions	63
Annex U (normative):	Extended graphics APIs	64
Annex V:	Void	65
Annex W (informative):	DVB-J examples.....	66
W.1	DVB-J examples from MHP	66
W.2	Example of enumeration extension	66
Annex X (normative):	Test support.....	68
Annex Y (normative):	Inter-application and Inter-Xlet communication API.....	69
Annex Z (informative):	Services, service contexts and applications in an MHP environment	70
History		71

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

All published ETSI deliverables shall include information which directs the reader to the above source of information.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECTrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

Founded in September 1993, the DVB Project is a market-led consortium of public and private sector organizations in the television industry. Its aim is to establish the framework for the introduction of MPEG-2 based digital television services. Now comprising over 200 organizations from more than 25 countries around the world, DVB fosters market-led systems, which meet the real needs, and economic circumstances, of the consumer electronics and the broadcast industry.

1 Scope

The present document defines the GEM platform based on MHP [1]. GEM is applicable for specifications and standards based on the MHP APIs, content formats, and semantic guarantees.

The present document is firstly intended to be used by entities writing terminal specifications and/or standards based on MHP. Secondly it is intended for developers of applications that use the GEM functionality and APIs. The GEM specification aims to ensure interoperability between GEM applications and different implementations of platforms supporting GEM applications. This includes interoperability across different middleware specifications, e.g. MHP, OCAP 1.0 [4], and ARIB AE [A]. Implementers should consult the publisher of specifications which reference GEM regarding conformance.

NOTE: The present document defines the interfaces visible to applications. Application developers should not assume that any related interface is available unless it is specifically listed. Terminal standards or implementations may have other interfaces present.

One of the primary goals of the present document is to minimize the number of divergences between MHP and terminal specifications based on GEM, wherever practical. divergence is defined in clause 3.1. Where divergences are inescapable, the present document serves as a place to document and control the permitted divergences, so that they will be predictable to terminal manufacturers, broadcasters, and application authors.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

- [1] ETSI TS 101 812: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.0.2".
- [2] ETSI TR 101 154 (V1.4.1): "Digital Video Broadcasting (DVB); Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in satellite, cable and terrestrial broadcasting applications".
- [3] ETSI TS 102 812: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1".
- [4] OCAP 1.0: "OpenCable Application Platform version 1.0".
- NOTE: <http://www.opencable.com/downloads/specs/OC-SP-OCAP1.0-I04-021028.pdf>.
- [5] ISO 639 (all parts): "Codes for the representation of names of languages".

3 Definitions and abbreviations

3.1 Definitions

3.1.1 Definitions from MHP

MHP [1], clause 11.3 is included in the present document, with the following notes and modifications.

In the body of definitions only, the interpretations described in clause 4.2 are to be applied.

3.1.2 Definitions introduced by GEM

For the purposes of the present document, the following terms and definitions apply:

divergence: everything that violates an assertion in a specification and/or a conformance clause

NOTE: A divergence from the MHP specification is when a correctly written conformance test for an MHP specification assertion would fail.

GEM application: application that is written only to the interfaces and semantic guarantees defined in GEM

NOTE: A suitably signalled GEM application will run on an MHP terminal, or on any terminal that complies to a GEM terminal specification, e.g. on OCAP and the ARIB AE.

functionally equivalent: functionally equivalent requirement is one that specifies behaviour that performs substantially the same function with substantially the same behaviour as the original specification, as seen from an application's point of view

NOTE: There are several clauses within TS 102 819 that do not require literal conformance with the corresponding requirement in the underlying MHP specification, but allow for a compatible substitution.

GEM terminal: terminal or other device that conforms to a GEM Terminal Specification

NOTE: Examples of GEM terminals include an MHP terminal, an OCAP terminal (including the POD) and a terminal supporting the ARIB AE.

GEM Terminal Specification: specification that includes all normative and selected optional elements of its underlying GEM document, and provides additional specifications that describe functionally equivalent elements for each and every clause of the underlying GEM document where required

standard definition: MPEG-2 main level at main profile, as defined in TR 101 154

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in MHP [1] and the following apply:

AIT	Application Information Table
API	Application Programming Interface
CA	Conditional Access
CLUT	Colour LookUp Table
DSMCC	Digital Storage Media Command and Control
DVB-J	DVB-Java
GEM	Globally Executable MHP
ID	IDentifier
IP	Internet Protocol
MHP	Multimedia Home Platform
NPT	Normal Play Time
NTSC	National Television Systems Committer
OCAP	OpenCable Applications Platform
POD	Point Of Deployment

SI	Service Information
SSL	Service Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UTF8	Universal Transformation Format 8
XML	eXtensible Markup Language

4 General considerations and conventions

4.1 General considerations

4.1.1 Purpose

The GEM document is not intended, and should not be used, as a complete terminal specification. It is a framework upon which a GEM terminal specification can be created.

The Multimedia Home Platform (MHP) middleware standard defines a comprehensive platform that enables interactive television services to be deployed that are interoperable across any manufacturer's implementations of the standard. MHP is a comprehensive specification of a receiving device (an MHP terminal). MHP terminals receive digital video broadcasting services based on MPEG-2 standards for various transmission media including satellite, cable, terrestrial and microwave. The transport layer may be DVB-T, DVB-C, or DVB-S.

One element of the MHP standard is a description of the terminal facilities that can be exploited by applications that form a part of a broadcast service. These facilities may be exposed via APIs (Application Programming Interfaces); such APIs carry semantic guarantees. Similarly, receiver functionality can be exposed with a declarative content format that contains semantic guarantees. Another element of the MHP standard is the specification of the terminal hardware and signalling infrastructure that allows it to be connected to any DVB-T, DVB-C or DVB-S network.

In some regions, markets and/or networks, it is impractical to adopt DVB-T, DVB-C or DVB-S signalling. For example, in the United States, there is a significant investment in infrastructure that cannot be easily converted. In Japan, the terrestrial broadcasting standard, while very similar to DVB-T, is not the same, and contains elements that make the adoption of the full MHP standard for terminals impractical.

Despite these regional differences, it is desirable to be able to execute a GEM application as part of a service that is carried over different network infrastructure. Such interoperability can be achieved, as long as the middleware standard supports the same APIs and semantic guarantees.

The present document for the Global Execution of MHP services (GEM) defines the APIs, semantic guarantees, and content formats that can be relied upon in all interactive television standards and specifications that support globally-interoperable MHP applications. Any such specification based on GEM shall normatively reference the GEM specification in its entirety, and shall fulfil the normative requirements of GEM.

The present document does not provide a complete specification sufficient to implement a device. Additional normative elements are required.

4.1.2 Format

The present document takes the form of a large number of normative references to the MHP specification. It does not invent new APIs or features; rather, it selects those portions of the MHP specification that define interfaces into terminal functionality. The GEM specification does not state how the receiver has to be built or what network infrastructure has to underlie the implementation; it is limited to specifying the behaviour and interfaces that globally interoperable applications may rely on.

This set of interfaces includes the vast majority of those that are defined in MHP. In certain rare cases, MHP contains APIs and/or other features that are inextricably tied to the specifics of the DVB network, e.g. the precise details of DVB service information. In these rare cases, it is impractical to require the behaviour specified by the MHP. In these cases, the appropriate elements of the MHP specification are explicitly called out as not being required by the GEM specification. In general, these features are not of interest to interoperable broadcast applications; they might be in MHP in support of other usage scenarios, such as an EPG provided by a network operator.

4.1.3 Inclusion of MHP features

4.1.3.1 Subsetting prohibited

Specifications that reference the present document has to include it in its entirety. It is prohibited to base any specification on the present document if the referencing document does not require all normative requirements of the present document.

4.1.3.2 Supersetting permitted

If a GEM terminal specification wishes to include APIs, signalling or behaviours defined in MHP [1] that are not required by GEM, it may do so by referring directly to MHP [1].

4.1.4 Addition of non-GEM interfaces

Terminal specifications based on GEM may add public interfaces, provided that they are added in a namespace that does not conflict with GEM. For example, OCAP 1.0 [4] defines extensions in the Java package `org.ocap`.

GEM terminal specifications and GEM terminals shall not require that such extension interfaces be called by GEM applications in order to enable behaviour that is normatively required by the present document.

4.1.4.1 DVB-J enumerations

A GEM terminal specification shall not add new values to an enumeration that is returned from a method defined by the present document.

NOTE: For example, the interface `org.dvb.net.rc.RCInterface` defined in annex R introduces an enumeration that is returned by the method `getType()`. This enumeration includes the values `TYPE_CATV`, `TYPE_DECT`, etc. It is not permissible to attempt to subdivide one of these types by introducing new enumeration values in a different namespace. See also the example in annex W.

4.1.5 Application areas

In this version of the GEM specification, the same application areas as MHP [1], clause 0.2 are considered.

4.1.6 Profiles

The informative text in MHP [1], clause 0.3 describes the MHP approach to profiles. The profiles defined in the present document are modelled on a similar scheme.

4.1.7 Full compliance with the present document

To be "fully compliant" with the present document, equipment shall also be fully compliant with any one of the following specifications.

- TS 101 812 [1] (MHP 1.0.X).
- TS 102 812 [3] (MHP 1.1.X).
- OCAP 1.0 [4].

For avoidance of doubt, equipment which is fully compliant with the entire present document apart from the above clause is not fully compliant with the present document.

4.2 Conventions

4.2.1 References within the MHP specification

MHP [1] contains numerous internal references. In certain cases, a clause of the MHP specification that is referenced by GEM will refer to a clause of the MHP specification that is not referenced by GEM, or to a clause whose requirements are modified by GEM. In the preparation of the GEM document, every effort has been made to identify these internal references, and indicate where they do not apply or where they should be interpreted as referring to a corresponding clause of GEM.

In case of error, such internal MHP references should be interpreted as referring to the appropriate clause of GEM. That is, if GEM modifies or removes a normative requirement of MHP, for the purposes of GEM any references to that clause of the MHP specification should be interpreted as referring to the appropriate clause of GEM.

4.2.2 Terminology in the MHP specification

4.2.2.1 MHP

The present document makes numerous references to MHP [1]. When a clause of the MHP specification is referenced from GEM, for the purposes of GEM references to MHP are to be interpreted to apply to GEM, and to terminal specifications based on GEM. Similarly, "MHP implementations" and "MHP terminal" are to be interpreted as "implementations of terminal specifications based on MHP," etc. "MHP application" is to be interpreted as "GEM application."

4.2.2.2 Resident navigator

MHP [1] uses the terms "navigator" and "resident navigator." It is noted that in terminal specifications based on GEM, it is permissible for some of the functions of the navigator to be delegated to an entity that is not part of the resident software of the terminal, e.g. the OCAP 1.0 [4] monitor application.

Downloaded or other resident applications that perform some of the policy decisions or functionality of the MHP navigator shall implement a policy that is consistent with the requirements of the present document.

4.2.2.3 DVB service

For the purposes of the present document, references within MHP [1] to DVB services shall be interpreted as meaning any services that carry GEM applications.

4.2.3 Inclusion of clauses of the MHP specification

Unless otherwise noted, inclusion of a chapter, annex or section of MHP [1] implies inclusion of all sub-sections.

5 Basic architecture

GEM does not mandate a basic architecture. Clause 5 of MHP [1] defines a basic architecture for MHP terminals. This is to be taken as an informative example of one possible architecture for terminal specifications based on GEM.

6 Transport protocols

6.1 Introduction

In order to be able to talk to the external world, a GEM terminal has to communicate through different network types.

Broadcast only services are provided on systems consisting of a downstream channel from the Service Providers to Service consumers.

Interactive services are provided on systems consisting of a downstream channel together with interaction channels.

6.2 Broadcast channel protocols

This clause deals with DVB defined or referenced broadcast channel protocols. This clause does not consider other protocols and the APIs that would provide access to them.

Other protocols and their APIs are considered as extensions to the present document, see annex H.

NOTE 1: Figure 8 in MHP [1], clause 6.2 shows the broadcast channel protocol stack for MHP. As some of the protocols are not required by the present document, not all elements of this figure necessarily apply; however terminal specifications based on GEM will need to define functional equivalents for any optional protocols they do not use.

The full details of APIs that provide DVB-J applications with access to broadcast protocols are in clause 9.

NOTE 2: MHP [1], clause 6.2 has a normative requirement related to conditional access descrambling and the section filter API. This is not a requirement of GEM.

6.2.1 MPEG-2 transport stream

MHP [1], clause 6.2.1 is included in the present document.

6.2.2 MPEG-2 clauses

MHP [1], clause 6.2.2 is included in the present document.

6.2.3 DSM-CC private data

MHP [1], clause 6.2.3 is included in the present document.

6.2.4 DSM-CC data carousel

MHP [1], clause 6.2.4 is included in the present document.

6.2.5 DSM-CC user-to-user object carousel

MHP [1], clause 6.2.5 is included in the present document, with the following notes and modifications. For this clause, sub-clauses are only included if this is explicitly indicated.

Terminal specifications based on GEM have to specify a signalling mechanism for the delivery of a hierarchical file system.

6.2.5.1 DVB-J class files

MHP [1], clause 6.2.5.1 is included in the present document, with the following notes and modifications.

If the terminal specification does not use the BIOP::FileMessage structure, then the equivalent mechanism for delivering a file shall be used to deliver a "class" file, as described in of MHP [1], clause 6.2.5.1.

6.2.5.2 DVB-HTML document files

Void.

6.2.5.3 Loss of carousel behaviour

MHP [1], clause 6.2.5.3 is included in the present document, with the following notes and modifications.

The conditions for permanent loss of a carousel may be specified differently from MHP in terminal specifications based on GEM, therefore the reference to MHP [1], clause B.2.11 need not necessarily apply. However, terminal specifications based on GEM shall specify conditions for permanent loss of a carousel. The present document does include MHP [1], clause 9.1, thus the conditions for temporary disconnection and reconnection of a carousel as defined in MHP [1], clause 9.1.5 do apply. Thus, the language in MHP [1], clause 6.2.5.3 following the first paragraph to apply to the present document.

6.2.6 Protocol for delivery of IP multicast over the broadcast channel

MHP [1], clause 6.2.6 "DVB Multiprotocol Encapsulation" is included in the present document, with the following notes and modifications.

Use of this protocol is not required for terminal specifications based on GEM, however some mechanism for delivery of IP multicast over the broadcast channel if support for IP over the broadcast channel is supported. This feature is optional in all profiles of the present document.

6.2.7 Internet Protocol (IP)

MHP [1], clause 6.2.7 is included in the present document.

6.2.8 User Datagram Protocol (UDP)

MHP [1], clause 6.2.8 is included in the present document.

6.2.9 DVB service information

MHP [1], clause 6.2.9 is *not* included in the present document, however, terminal specifications in GEM shall provide a mechanism for delivery of service information that is sufficient for the SI access mechanisms required by GEM.

6.3 Interaction channel protocols

MHP [1], clause 6.3 is included in the present document, with the following notes and modifications.

Unless explicitly noted below, the listed protocols are not mandated in any profile in the present document. Non-required protocols are included in the present document for informative purposes, and to provide definitions.

GEM terminals that support IP shall be compatible with Internet Protocol as defined in MHP [1], clause 6.3.2.

GEM terminals that support TCP shall be compatible with Transmission Control Protocol as defined in MHP [1], clause 6.3.3.

GEM terminals that support UDP shall be compatible with UDP as defined in MHP [1], clause 6.3.9.

7 Content formats

7.1 Static formats

MHP [1], clause 7.1 is included in the present document.

7.2 Broadcast streaming formats

7.2.1 Audio

At least one format for streaming audio has to be specified in a GEM terminal specification.

7.2.2 Video

At least one format for delivering standard definition streaming video has to be specified in a GEM terminal specification.

7.2.3 Subtitles

Support for DVB subtitles as specified in MHP [1], clause 7.2.3 is optional in the present document.

NOTE: OCAP 1.0 [4] does not include support for subtitles. It does include US closed-captioning which is somewhat similar, but has different regulatory requirements and usage models.

7.3 Resident fonts

MHP [1], clause 7.3 is included in the present document.

7.4 Downloadable fonts

MHP [1], clause 7.4 is included in the present document.

7.5 Colour representation

MHP [1], clause 7.5 is included in the present document.

7.6 MIME types

MHP [1], clause 11.5 is included in the present document, with the following notes and modifications.

NOTE: The entries for "image/dvb.subtitle", "text/dvb.subtitle", "text/dvb.teletext" and "multipart/dvb.service" refer to content types for which support is not required by the present document.

8 DVB-HTML

The GEM specification provides the basic definitions needed for integration of DVB-HTML applications into a subsequent version of GEM:

- Definition of the term "DVB-HTML application," from MHP [1], clause 3.1.
- A framework of requirements on the signalling of applications that can be extended to support DVB-HTML in the future.

A definition of the content and application format from the HTML family is not in this version of the present document.

9 Application model

9.1 Broadcast GEM applications

MHP [1], clause 9.1 is included in the present document, with the following notes and modifications.

In this clause, the terms "AIT" and "application descriptor" are to be interpreted as referring to the application description defined in clause 10.4. The term "DVB service" is to be interpreted as meaning "service." Additionally, attention is drawn to the general rules in clause 4.2.

Support for host control tune requests is not mandatory in the present document, thus the language in the first paragraph of clause 9.1.1 relating to these tune requests only applies if such control is present in the terminal specification.

In clause 9.1.5, the reference to MHP [1], clause 6.2.5.3 is to be interpreted as referring to clause 6.2.5.3. The language at the end of MHP [1], clause 9.1.5 relating to the PMT information only applies to terminal specifications that feature this signalling.

9.2 DVB-J model

MHP [1], clause 9.2 is included in the present document, with the following notes and modifications.

The reference to the `application_control_code` parameter of the AIT in MHP [1], clause 9.2.3.2 is to be interpreted as referring to the `application_control_code` defined in clause 10.4.

9.3 DVB-HTML model

MHP [1], clause 9.3 is included in the present document, with the following notes and modifications.

In MHP [1], clause 9.3.2.2, the reference to clause 10 pertaining to the signalling of an HTML application does not apply. An abstract model for the signalling of an HTML application will be defined in a future version of the present document. All references to signalling in clauses 9.3.2.2 and 9.3.2.3 are to be read as referring to this abstract model.

9.4 Inter-application resource management

MHP [1], clause 9.4 is included in the present document, with the following notes and modifications.

The reference to the `application_priority` field in the application descriptor is to be interpreted as referring to the `application_priority` defined in clause 10.4.

Some downloaded resident applications specified as extensions to the present document may perform some of the functions of the MHP navigator, e.g. the monitor application defined OCAP 1.0 [4]. In this case, such downloaded software has to implement a policy that is consistent with the requirements of the present document, e.g., MHP [1], clause 9.4.

10 Application signalling

10.1 Introduction

This clause covers the following topics:

- Identification and launching of applications associated with a service.
- Requirements on the signalling that enables a broadcast to manage the lifecycle of applications.

MHP [1] contains a model of signalling that fulfils the requirements of GEM, but other signalling is possible. Broadly speaking, GEM places requirements on both the format of an application and requirements underlying its signalling. GEM does not, however, define the signalling that has to be used or the packaging of applications; this is left for GEM-based specifications to define.

10.1.1 Summary of requirements on common signalling

The minimum signalling requirements for any GEM application are summarized as follows:

- Some form of Application Description 10.4 with information sufficient to:
 - identify the source of the application code and other assets;
 - identify the application's application ID and organization ID;
 - identify the name of the application.

10.1.2 Summary of additional signalling for DVB-J applications

The minimum additional signalling requirements for DVB-J applications are summarized as follows:

- A DVB-J Specific Application Description 10.5 with information sufficient to:
 - signal parameters to the application;
 - indicate the initial class of the application.

10.2 Program specific information

A service carrying GEM applications has to contain information sufficient to locate the following:

- the Application Description 10.4 for each application in the service;
- the source of the application code and data.

10.3 Locators within an application description

Some fields of the application description contain locators, e.g. locators to a directory containing certain kinds of files. These locators can be to any transport defined within a GEM terminal specification, e.g. they can be locators to an object carousel, part of a data carousel, an http URL, etc. GEM does not mandate any particular transport. It does, however, require at least one transport that is capable of carrying the information needed to launch applications. This transport has to be capable of carrying files, or directory hierarchies containing files. The ability to list the contents of a directory is optional.

10.4 Application description

The Application Description provides full information on an application, its parameterization, the required activation state of it etc. Specifications based on GEM have to permit the signalling of multiple applications per service, without any arbitrary upper bound less than 255.

Data in the Application Description allows the broadcaster to request that the GEM terminal change the activation state of an application.

MHP [1], clause 10.4 defines an Application Information Table that fulfils this requirement.

10.4.1 Application description transmission and monitoring

It shall be possible to arrange for signalling such that the maximum time interval between the moment the application description is updated and the moment the new version is detected by the terminal will be no more than 30 s.

10.4.2 Visibility of application description

If an application tunes away from a transport stream where its signalling is carried without selecting a new service, it shall be permitted to continue running even if the application description is no longer available to the GEM terminal. For example, MHP [1], clause 10.4.4 defines behaviour consistent with this requirement.

10.4.3 Content of the application description

The Application Description describes applications and their associated information. It has to contain information sufficient to derive the following:

Table 1: Application description

Function	Type
application_type	enumeration
organization_id	32 bit unsigned integer
application_id	16 bit unsigned integer
application_control_code	enumeration
application_profiles_count	4 bit unsigned integer
for (i=0; i<N1; i++) {	
application_profile	16 bit unsigned integer
version.major	8 bit unsigned integer
version.minor	8 bit unsigned integer
version.micro	8 bit unsigned integer
}	
service_bound_flag	boolean
visibility	enumeration
application_priority	8 bit unsigned integer
application_name	String
application_icon_locator_count	unsigned integer
for (i=0; i<N2; i++) {	
application_icon_locator	Locator
application_icon_flags	16 bit unsigned integer
}	

application_type: Identifies the type of application. Specifications based on GEM shall provide a mechanism for indicating those application types defined in MHP [1], clause 10.4.6, e.g. DVB-J and DVB-HTML.

organization_id: An organization_id, as defined in MHP [1], clause 10.5.1 under organization_id. In GEM, inclusion of this value in the "leaf" certificate of an authenticated application is required, as it is in MHP.

application_id: An application_id, as defined in MHP [1], clause 10.5.1 under application_id.

application_control_code: An application control code, as defined in MHP [1], clause 10.6.2.1. Support for the REMOTE application type is not required, but may optionally be present in terminal specifications based on GEM.

application_profile: Information sufficient to derive the MHP profile on which this application could execute, as defined in MHP [1], clause 10.7.3.

application_profiles_count: The number of application profiles signalled for this application.

version.major: The major sub-field of the profile version number, as defined in MHP [1], clause 10.7.3.

version.minor: The minor sub-field of the profile version number, as defined in MHP [1], clause 10.7.3.

version.micro: The micro sub-field of the profile version number, as defined in MHP [1], clause 10.7.3.

The four fields above indicate the minimum MHP profile on which an application will run. For example, an application that relies on the guarantees of GEM 1.0 would run on an appropriate profile of MHP 1.0.2. The underlying signalling of the application has to indicate the minimum profile that the application requires in a way that can be mapped to MHP profiles and the MHP version number.

service_bound_flag: A service bound flag, as defined in MHP [1], clause 10.7.3. Terminal specifications based on GEM are required to support the MHP semantics of this field.

visibility: A visibility field, as defined in MHP [1], clause 10.7.3.

application_priority: An application priority, as defined in MHP [1] 10.7.3. Terminal specifications based on GEM have to support at least 32 priority levels, with the semantics spelled out in MHP's definition of this value.

application_name: A string that names the application in a way meant to be informative to the user. The signalling shall support strings whose UTF8 encoding is up to 128 bytes, not including any termination character. It is permissible to signal more than one application name, e.g. the application name could be given in several different languages, with a method for determining which one is to be presented to the user, as is done in MHP. In all cases, it shall be possible to associate an ISO 639 [5] language code with each application name. It shall be possible to signal any string that can be represented with UTF8.

application_icon_locator_count: The number of application icon locators associated with this application. Signalling to support values of 0 and 1 shall be present. Terminal specifications based on GEM may support any number of application icon locators.

application_icon_locator: Information sufficient to derive a locator to a directory containing application icons. The application icons shall be in files in the directory indicated by this locator, in the format specified in MHP [1], clause 10.7.4.2.

application_icon_flags: Flags describing the icon files in the directory identified by the application_icon_locator, in the format specified in MHP [1], clause 10.7.4.

10.4.4 Applications from previously selected services

If an application with a service_bound_flag of 0 is running when a service selection is performed, it shall continue to run in a newly selected service if the same application is signalled in the new service. To efficiently support this feature on services that do not contain the application code, it may be desirable to have signalling equivalent to that described in MHP [1], clause 10.7.5.

10.5 DVB-J specific application description

10.5.1 General

Additional signalling specific to DVB-J applications has to be present in terminal specifications based on GEM.

10.5.2 Content of DVB-J application description

For each application description that refers to a DVB-J application, it has to be possible to signal information sufficient to derive the following:

Table 2: DVB-J application description

Function	Type
for (i=0; i<N; i++) { dvbj_app_parameter }	String
base_directory	Locator
for (i=0; i<N; i++) { classpath_element (optional) }	Locator
initial_class_name	String

dvbj_app_parameter: A string that is passed to the application as parameters. The signalling shall support parameter strings such that a minimum total length of 240 bytes can be supported. The length is calculated as the sum of (1 + the sum of (1 + length(dvbj_app_parameter))) where the length of a parameter is the length of that parameter string encoded in UTF8, with no termination character. It shall be possible to signal any string that can be represented with UTF8.

NOTE: MHP [1] exceeds this requirement somewhat; see MHP [1], clause 10.9.1.

initial_class_name: The fully-qualified name of the initial class of this application. This class has to implement the Xlet interface. The signalling has to support UTF8 encoding up to 80 bytes, not including any termination character. It shall be possible to signal any string that can be represented with UTF8.

base_directory: A locator specifying a directory. This directory is used as a base directory for relative path names. This base directory is automatically considered to form the first directory in the class path (after the path to the system's classes).

classpath_element: GEM-based terminal specification may include optional signalling to indicate a list of other locators to be added to an application's class path. For example, MHP [1], clause 10.9.2 defines the classpath_extension for this purpose. If support for this is included in a terminal specification, there may be restrictions placed on these locators, e.g. that they represent sub-directories of the base_directory.

11 DVB-J platform

11.1 The virtual machine

MHP [1], clause 11.1 is included in the present document.

11.2 General issues

MHP [1], clause 11.2 is included in the present document.

11.3 Fundamental DVB-J APIs

MHP [1], clause 11.3 is included in the present document, with the following notes and modifications.

NOTE 1: MHP [1], clause 11.3.1.1 bullet point g does apply to the present document. Thus, all terminal specifications based on GEM require support for the system property "dvb.persistent.root".

MHP [1], clause 11.3.1.6 includes a definition for the behaviour of `URL.getContent()`. Part of this definition is a priority for the data type of the URL, including the content type descriptor in an object carousel. If a GEM terminal specification does not include support for an object carousel, this requirement obviously would not apply; however, if the equivalent signalling contains data type information, it is recommended that it be given the same priority as the content type descriptor is given in MHP.

MHP [1], clause 11.3.2.1 contains a reference to the class `org.davic.net.dvb.DVBLocator`. This class is not required by GEM. This is to be interpreted as allowing a valid locator as described, where that locator is formed as described below.

The present document does not require a particular text encoding for locators, however terminal specifications are required to define such a text encoding. The entities for which a text encoding is required are specified in clause 14.8.

Where a locator text encoding is required, a locator may be constructed from the text representation using the factory method defined in the class `javax.tv.locator.LocatorFactory`.

NOTE 2: Portable GEM applications should not contain hard-coded text representations for locators, as it is likely that the locators will vary across networks. If an application needs to be signalled with values for locators, they can be passed in as Xlet arguments, or put in a small text file that is read from the carousel.

11.4 Presentation APIs

MHP [1], clause 11.4 is included in the present document, with the following notes and modifications.

NOTE 1: As a consequence of clause 4.1.4, the requirements of MHP [1], clause 11.4.1.4 are required of all GEM-based terminal specifications; as a further consequence, terminal specifications based on GEM shall not define extensions that have to be invoked by applications in order to obtain the behaviour mandated by the present document.

NOTE 2: MHP [1], clause 11.4.1.4 contains a requirement that applications cover at least 3% of the visible area on the screen under certain circumstances. Obviously, the pixel values given only apply to systems with the standard definition resolution required by MHP; on other systems, the 3% requirement applies, but results in different pixel values.

The last paragraph of MHP [1], clause 11.4.2.2 places a requirement on the handling of DVBLocators. As DVBLocator is not required by the present document, this paragraph does not apply. Instead, the present document requires that any information in a locator beyond that identifying a service (e.g. the time of a specific program event) is to be ignored by JMF players. See also clause 14.8.

MHP [1], clause 11.4.2.5.1 requires the following classes:

- `org.davic.media.SubtitlingLanguageControl`
- `org.dvb.media.SubtitlingEventControl`
- `org.dvb.media.SubtitleAvailableEvent`
- `org.dvb.media.SubtitleListener`
- `org.dvb.media.SubtitleNotAvailableEvent`
- `org.dvb.media.SubtitleNotSelectedEvent`
- `org.dvb.media.SubtitleSelectedEvent`
- `org.dvb.media.CAStopEvent`
- `org.dvb.media.CAException`

These classes are not required to be present by the present document.

MHP [1], clauses 11.4.2.5.4 and 11.4.2.5.5 refer to classes not required by the present document. These references are not a part of the present document.

MHP [1], clause 11.4.2.7 refers to the component tags of a locator. For the purposes of GEM, this is to be interpreted as meaning the description of the required components in a locator.

11.5 Data access APIs

MHP [1], clause 11.5 is included in the present document, with the following notes and modifications.

The reference to annex P in MHP [1], clause 11.5.1 is to be read as referring to annex P of the present document.

The reference to annex R in MHP [1], clause 11.5.5 is to be read as referring to annex R of the present document.

11.6 Service information and selection APIs

11.6.1 DVB service information API

The DVB specific SI API is **not** required in the present document. Thus, MHP [1], clause 11.6.1 is not considered to be included in the present document.

11.6.2 Service selection API

MHP [1], clause 11.6.2 is included in the present document.

11.6.3 Tuning API

MHP [1], clause 11.6.3 is included in the present document, with the following notes and modifications.

The reference to the `DvbLocator` class does not apply to the present document. The reference to MHP [1], clause 1.7.6 is to be read as referring to clause 11.7.6.

11.6.4 Conditional access API

The present document does not require a conditional access subsystem, nor does it place requirements on one if it is present. Thus, MHP [1], clause 11.6.4 is not included in the present document.

11.6.5 Protocol independent SI API

MHP [1], clause 11.6.5 is included in the present document, with the following notes and modifications.

The mapping of the protocol independent SI API onto the underlying SI protocol is not defined in the present document. Thus, the reference to MHP [1], annex O does not apply. However, GEM terminal specification has to provide a mapping of the protocol independent SI API onto their SI signalling.

11.7 Common infrastructure APIs

11.7.1 APIs to support DVB-J application lifecycle

MHP [1], clause 11.7.1 is included in the present document, with the following notes and modifications.

NOTE: The xlet properties "dvb.org.id", "dvb.app.id" and "dvb.caller.parameters" have to be supported.

In MHP [1], clause 11.7.1.1, the reference to the DVB-J application descriptor is to be interpreted as referring to "Content of DVB-J application description". The text requiring that a specific text encoding be used does not apply to the present document.

11.7.2 Application discovery and launching APIs

This API is formed of the `org.dvb.application` package defined in annex P.

NOTE 1: This is the same API as in MHP [1].

The following properties are defined for use with the method `AppAttributes.getProperty`:

Table 3: Application attribute properties

Property name (see note)	Return
<code>dvb.j.location.base</code>	Returns String containing <code>base_directory</code> from the DVB-J application description.
<code>dvb.j.location.cpath.extension</code>	Returns <code>String[]</code> derived from <code>classpath_element</code> from the DVB-J application description with each array entry corresponding to a pathname entry as defined for <code>classpath_element</code> .
NOTE: Property names beginning "dvb." are reserved for future use.	

NOTE 2: The property `dvb.transport.oc.component.tag` is not required by the present document.

The following table defines the source of the information which shall be used for methods returning information from entries in the application database for an application signalled in an application description.

Table 4: Information source for methods on AppAttributes

Method	Information source
getName()	One of the names that can be found in the application_name of the Application description.
getName(String ISO639code)	A name of the application_name of the Application description corresponding to the specified language, if available.
getNames()	All of the names for the application which can be found in the application_name of the Application description and their ISO 639 language code.
getProfiles()	The set of profiles indicated in the application_profile of the Application description.
getPriority()	The value indicated for the application_priority of the Application description.
getVersions(String profile)	The values version.major, version.minor and version.micro for the specified profile from the Application description.
getIsServiceBound()	True if the service_bound_flag of the Application description indicates true. Otherwise false.
isStartable()	There is no information source for this method, the return value is derived as specified in the method description. For the purpose of the method description, remote applications are as specified in the GEM terminal specification, if they are supported.
getIdentifier()	The organization_id and application_id of the Application description.
getServiceLocator()	If remote applications are supported, the locator for a remote application shall encapsulate the values found in the appropriate signalling in the terminal specification.
getLocator()	The application_icon_locator of the Application description.
getIconFlags()	The application_icon_flags of the Application description.

11.7.3 Inter-application communication API

MHP [1], clause 11.7.3 is included in the present document.

11.7.4 Basic MPEG concepts

MHP [1], clause 11.7.4 is included in the present document, with the following notes and modifications.

The classes `DvbElementaryStream`, `DvbService`, and `DvbTransportStream` are not required by the present document. The note requiring the return of the DVB specific subclass for methods returning instances of elementary stream, service or transport stream does not apply to the present document.

11.7.5 Resource notification

MHP [1], clause 11.7.5 is included in the present document.

11.7.6 Content referencing

This API is formed of the `DAVIC Locator` class and the `javax.tv.locator` package, both as described in MHP [1], clause 11.7.6. The `DAVIC` class `DvbLocator` is not required by the present document.

The signature of the `org.davic.net.Locator` class shall be extended with:

- `"implements javax.tv.locator.Locator"`.

The `createFactory()` method of `javax.tv.locator.LocatorFactory` shall always return `org.davic.net.Locator(s)` which implement the `javax.tv.locator.Locator` interface when provided with a locator syntax that is valid in the terminal specification. See also clause 14.8.

In the present document, methods whose signature has a return type of `org.davic.net.Locator` or `javax.tv.locator.Locator` shall return an instance of `org.davic.net.Locator` (or a subclass of that) where the locator returned can be represented by the locator syntax described by the terminal specification. In this case, the locator returned shall contain an identification of a service.

Any optional extensions of locators (e.g. for specifying components, events etc.) are considered in a comparison and if they are not equally present in both locators then the comparison shall fail.

For the above locators "best effort" comparison shall be exact.

The protected constructor of `LocatorFactory` is for implementation use. MHP applications shall not subclass `LocatorFactory`. Implementations are not required to behave correctly if they should do this.

11.7.7 Common error reporting

MHP [1], clause 11.7.7 is included in the present document, with the following notes and modifications.

The interface `NotAuthorizedInterface` and the exception `NotAuthorizedException` are not required by the present document.

11.8 Security

MHP [1], clause 11.8 is included in the present document.

11.9 Other APIs

11.9.1 Timer support

MHP [1], clause 11.9.1 is included in the present document.

NOTE: The minimum repeat interval of 40ms in MHP was motivated by a standard definition frame rate of 25 Hz, however this was not meant to imply that the timer could be used for frame-accurate animation.

11.9.2 User settings and preferences API

MHP [1], clause 11.9.2 is included in the present document.

11.9.3 Profile and version properties

MHP [1], clause 11.9.3 is included in the present document, with the following notes and modifications.

All of the system properties defined in this clause is required by the present document. For GEM-based terminal specifications, the properties indicating the profile (`MHP.profile.enhanced_broadcast`, `MHP.profile.interactive_broadcast` and `MHP.profile.internet_access`) are to be interpreted as referring to the profile descriptions in clause 15 of the present document. The properties referring to version numbers are to be interpreted as referring to the corresponding *MHP* version number.

NOTE: This means that a receiver implementing a terminal specification based on the interactive broadcast profile of GEM 1.0 would return property values consistent with MHP 1.0.2's enhanced broadcast profile.

11.10 Java permissions

MHP [1], clause 11.10 is included in the present document, with the following notes and modifications.

In MHP, the only mechanism by which an application may be trusted, and thus request additional permissions, is for that application to be signed. As described in clause 12.1.3, terminal specifications based on GEM may add other mechanisms for establishing that an application is trusted. Thus, in this clause the term "signed application" is to be interpreted as meaning an application that is eligible for being granted permissions beyond the MHP sandbox. "Unsigned application" is to be interpreted as meaning an application that has not been packaged in such a way.

Any additional mechanism complementing the MHP codesigning model for granting trust to an application shall be defined in a GEM-based terminal specification.

MHP [1], clause 11.10.2.2 refers to object carousels. This is to be interpreted as meaning object carousels, or any other filesystem that may be mounted using the DSMCC APIs.

The class `org.dvb.net.ca.CAPermission` is not required by the present document; MHP [1], clause 11.10.2.3 does not apply for GEM terminals where this class is not present.

11.11 Content referencing

11.11.0 General

The following mapping shall be used between the types of locator defined in table 5 and the DVB-J methods defined in this clause. It lists the Java methods and constructors that accept or return (as defined by their method signature) instances of `org.davic.net.Locator`, `javax.tv.locator.Locator`, `javax.media.MediaLocator` or their subclasses. The external form of the locators shall as described in table 5 for the corresponding entity being referenced. Where the same method is listed as accepting multiple forms of locator, then it is required to accept all forms listed in this clause.

Where a method listed below is defined (in its specification) to check its input then it shall only accept the forms of locator listed below as being valid for that method from among those defined in the present document. Other forms of locator from among those defined in the present document shall be rejected as specified for the method concerned. If a method does not specify a means of rejecting inappropriate locators then it shall fail silently apart from Exceptions and Events which do not check their input and where it is the responsibility of the platform to use correct locators when constructing them. The present document does not prevent methods accepting other forms of locator that are not defined in the present document.

11.11.1 Transport stream

MHP [1], clause 11.11.1 is included in the present document, with the following notes and modifications.

The term "DVB locators" is considered to refer to all valid locators as described in table 5.

11.11.2 Network

MHP [1], clause 11.11.2 is included in the present document, with the following notes and modifications.

The term "DVB network" is to be interpreted as referring to a valid network, as described in table 5.

11.11.3 Bouquet

MHP [1], clause 11.11.3 is not included in the present document.

11.11.4 Service

11.11.4.1 MPEG/GEM specific service

MHP [1], clause 11.11.4.1 is included in the present document, with the following notes and modifications.

The term "DVB service" is to be interpreted as meaning "GEM service". "DVB locator" is to be interpreted as meaning "GEM locator."

The following methods are *not* required by the present document:

- `org.davic.net.ca.CAModule.buyEntitlement()`
- `org.davic.net.ca.CAModule.queryEntitlement()`
- `org.dvb.si.SIDatabase.retrieveSIService()`
- `org.dvb.si.SIDatabase.retrievePMTService()`
- `org.dvb.si.PMTService.getDvbLocator()`
- `org.dvb.si.SIBouquet.getSIServiceLocators()`
- `org.dvb.si.SIService.getDvbLocator()`
- `org.davic.net.ca.TuneRequestEvent` - constructor
- `org.davic.net.ca.TuneRequestEvent.getLocator()`

11.11.4.2 Generic service

MHP [1], clause 11.11.4.1 is included in the present document, with the following notes and modifications.

The term "DVB specific service" is to be interpreted as meaning "GEM service."

11.11.5 Program event

MHP [1], clause 11.11.5 is included in the present document, with the following notes and modifications.

The term "DVB Event" is to be interpreted as meaning "program event".

The following methods are *not* required by the present document:

- `org.davic.net.ca.CAModule.buyEntitlement()`
- `org.davic.net.ca.CAModule.queryEntitlement()`
- `org.dvb.si.SIEvent.getDvbLocator()`

11.11.6 MPEG elementary stream

MHP [1], clause 11.11.6 is included in the present document, with the following notes and modifications.

The phrase "DVB locators including multiple component tags" is to be interpreted as meaning "GEM locators including a reference to multiple components." In the bulleted list, the note "shall also accept multiple component tag "dvb:" locator" shall be interpreted as referring to these same GEM locators.

The following methods are *not* required by the present document:

- `org.dvb.si.SIDatabase.retrievePMTElementaryStreams()`
- `org.dvb.si.PMTElementaryStream.getDvbLocator()`
- `org.davic.net.ca.DescramblingStoppedEvent.getServiceLocator()`
- `org.davic.net.ca.DescramblingStartedEvent.getServiceLocator()`

11.11.7 File

MHP [1], clause 11.11.7 is included in the present document, with the following notes and modifications.

The note about "instances of "dvb:" locator including dvb_abs_path" shall be interpreted as meaning locators referring to File or Directory entities, as defined in table 5.

11.11.8 Directory

MHP [1], clause 11.11.8 is included in the present document, with the following notes and modifications.

The phrase ""dvb:" locator" shall be interpreted as meaning GEM locator.

11.11.9 Drip feed decoder

MHP [1], clause 11.11.9 is included in the present document.

11.11.10 Irrelevant

MHP [1], clause 11.11.10 is included in the present document.

11.11.11 Methods working on many locator types

The following methods used in the present document work on many locator types. The locator types which each method is required to support are listed for each of the methods concerned.

- `javax.tv.locator.LocatorFactory.transformLocator` - transforms a transport independent locator into a transport dependent one:
 - required to accept instances of `org.davic.net.Locator` describing a transport independent service;
 - required to return instances of `org.davic.net.Locator` describing a transport dependent service.
- `javax.tv.locator.LocatorFactory.createLocator` - creates a locator from a string:
 - required to accept valid GEM locators (see clause 14.8) and return corresponding instances of `org.davic.net.Locator`.
- `javax.tv.service.SIManager.registerInterest` - accepts a locator referencing one or more `SIElements` as input.
- `javax.tv.service.SIManager.retrieveSIElement` - accepts a locator referencing one or more `SIElements` as input:
 - both these methods are required to accept locators referencing: -Bouquet, Network, Event, ElementaryStream, Service, TransportStream.
- `javax.tv.service.SIElement.getLocator`:
 - returns a locator for "this `SIElement`" as specified by the JavaTV specified sub-interfaces.

11.11.12 Support for the HTTP Protocol in DVB-J

MHP [1], clause 11.11.12 is included in the present document.

12 Security

12.1 Introduction

This clause covers the following areas of security:

- Authentication of applications.
- Security policies for applications.
- Authentication and privacy of the return channel communications.
- Certificate management.

12.1.1 Overview of the security framework for applications

MHP [1], clause 12.1.1 is included in the present document, with the following comments and modifications.

In the last paragraph of clause 12.1.1, "applications that are signed" is to be interpreted as referring to any application that is eligible to be trusted, either through the MHP mechanism or through other mechanisms, as specified in the present document in clause 12.1.3. "unsigned applications" is to be interpreted as referring to applications that are not eligible to be trusted, through the MHP mechanism or other mechanisms.

12.1.2 Overview of return channel security

MHP [1], clause 12.1.2 is included in the present document.

12.1.3 Extensions to MHP application signing framework

In MHP, the only mechanism by which an application may be trusted, and thus request additional permissions, is for that application to be signed. Terminal specifications based on GEM may introduce additional mechanisms for establishing that an application is trusted. These mechanisms may involve some form of codesigning.

Any such extensions to the MHP security framework, whether they involve codesigning or not, have to:

- Require that trusted applications be identified with an `application_id` from the signed applications range, as described in clause 12.1.1.
- Refuse to grant permissions outside of the set granted to unsigned applications in MHP, unless those permissions are explicitly requested in the signalling.
- Must be completely specified in the GEM terminal specification.

NOTE 1: This requirement could be minimally satisfied with a flag in the signalling requesting that a large set of permissions be granted; it is recommended, however, that any extensions to the MHP security model feature some mechanism for enumerating permission requests at a level of detail comparable to MHP's.

NOTE 2: If this extension to the security framework involves a permission request file, we recommend that the permission request file starts with a name that identifies the organization defining the syntax, e.g. "ocap". Any extension mechanism has not to conflict with the MHP mechanism.

12.2 Authentication of applications

MHP [1], clause 12.2 is included in the present document.

12.3 Message transport

MHP [1], clause 12.3 is included in the present document.

12.4 Detail of application authentication messages

MHP [1], clause 12.4 is included in the present document.

NOTE: The exact file names, locations and syntaxes described in this clause has to be supported. This includes the requirement in clause 12.4.3.1 that the last certificate in a CertificateFile be the root certificate.

12.5 Profile of X.509 certificates for authentication of applications

MHP [1], clause 12.5 is included in the present document.

12.6 Security policy for applications

MHP [1], clause 12.6 is included in the present document, with the following notes and modifications.

As described in clause 12.1.3, mechanisms other than MHP codesigning may be used to determine if additional permissions should be granted to applications. As a consequence, in clause 12.4 the term "signed application" is to be interpreted as meaning an application that has been packaged in such a way that it is eligible for being granted additional permissions, either via the MHP signing mechanisms or through other mechanisms. "Unsigned application" is to be interpreted as meaning an application that has not.

NOTE 1: The policy for granting of permissions outlined in MHP [1], clause 12.6.1 is required to be supported. It is possible that these policy decisions will be made by an element that is downloaded to the terminal, e.g. OCAP 1.0 [4] defines a "monitor application" that makes policy decisions. In cases such as this, the downloaded element is required to implement a policy consistent with the present document.

NOTE 2: The exact syntax of the permission request file specified in MHP [1], clause 12.6.2 is required to be supported. Because of the rules in MHP [1], clause 14.3, it cannot be extended by adding tag definitions.

NOTE 3: If a terminal cannot support functionality implied by a permission tag, it has to still support the syntax of the permission tag. E.g. capermission tag has to be supported, even if support for the MHP CA APIs is present.

Table 54 in MHP [1], clause 12.6.2.4 contains a reference to the name `initial_class_byte`. For the present document, this is to be interpreted as referring to `initial_class_name` from table 2.

MHP [1], clause 12.6.2.8 discusses the permissions for conditional access. As discussed in clause 11.8, the class CAPermission is not required to be present; if it is not present, then the policies specified in clauses 12.6.2.8.1 and 12.6.2.8.2. However, the syntax of the `capermission` tag as specified in clause 12.6.2.8.3 has to be supported in all cases; if the CAPermission class is not present, it is to be silently ignored.

MHP [1], clause 12.6.2.9 refers to the AIT of a service. For the present document, this is to be interpreted as referring to the application description, as described in clause 10.4.

NOTE 4: The return channel access policy and permission described in MHP [1] are required to be supported. Attention is drawn to the note at the end of clause 12.6.1 relating to the return channel permission and return channel connections where it is not necessary to dial a phone, e.g. cable modems.

12.7 Example of creating an application that can be authenticated

MHP [1], clause 12.7 is included in the present document.

12.8 GEM/MHP certification procedures

Certification procedures are outside the scope of the present document.

12.9 Certificate management

MHP [1], clause 12.9 is included in the present document.

12.10 Security on the return channel

MHP [1], clause 12.10 is included in the present document.

12.11 The internet profile of X.509 (informative)

MHP [1], clause 12.11 is included in the present document.

12.12 Platform minima

MHP [1], clause 12.12 is included in the present document.

13 Graphics reference model

13.0 General

MHP [1], clause 13 is included in the present document, with the following notes and modifications.

13.1 Supported graphics resolutions

MHP [1], clause 13 contains references to the platform minima in clause G, e.g. in clauses 13.2.1.3 and 13.3.6.1. For the present document, these references are to be interpreted as referring to annex G.

Table 62 in MHP [1], clause 13.2.1.3 is an informative listing of typical resolutions and their pixel aspect ratios. This may not apply in all regions, e.g. regions with NTSC standard definition.

13.2 Broadcast streaming formats

MHP [1], clause 13.4.1 mandates background players for the broadcast streaming formats. The present document does not mandate a particular broadcast streaming format, e.g. Standard Definition 25Hz MPEG-2 Video is not required by the present document. A GEM terminal specification has to include some mechanism for delivering MPEG-2 audio and video programming in standard definition, however. For these formats, background JMF players have to be created. Thus, the last paragraph of clause 13.4.1 applies to this player.

13.3 Subtitles

As signalling to support subtitles is not required by the present document, MHP [1], clause 13.5 only applies to terminal specifications based on GEM for which subtitling is available. If subtitling signalling is available, the presentation of subtitles has to follow the model specified in MHP [1], clause 13.5.

NOTE: US closed captioning is not the same thing as subtitles, thus, in systems where closed captioning is available but subtitling is not, clause 13.5 is considered to be optional.

14 System integration aspects

14.1 Namespace mapping

The present document does not mandate any particular format for locators. Note, however, that 14.8 requires that terminal specifications based on the present document define some text representation for certain entities.

14.2 Reserved names

MHP [1], clause 14.2 is included in the present document.

14.3 XML notation

MHP [1], clause 14.3 is included in the present document, with the following notes and modifications.

These XML notation rules only apply to XML file formats defined in the present document, or in MHP [1].

In the fourth bullet item, MHP prohibits indicating an encoding attribute in an XMLDecl item to specify an encoding other than UTF-8. The present document relaxes this prohibition: Terminal specifications based on the present document may extend the allowed XML notation by permitting other character encodings. If no encoding is specified, however, the default shall be UTF-8.

14.4 Network signalling

The present document does not mandate specific network signalling, nor error behaviour when incorrectly formatted data is received.

14.5 Text encoding of application identifiers

MHP [1], clause 14.5 is included in the present document, with the following notes and modifications.

The references to `organization_id` and `application_id` are to be interpreted as referring to the versions of `organization_id` and `application_id` from the present document.

14.6 Reserved names for persistent storage

MHP [1], clause 14.6 is included in the present document.

14.7 Files and file names

MHP [1], clause 14.8 is included in the present document, with the following notes and modifications.

The reference to "using a DVB locator including the `dvb_abs_path` part of the name part of the syntax" shall be interpreted to mean the use of a locator that refers to a file or directory, as described in clause 14.8.

The locator format has to be able to reference any valid file name; at least the minima specified in clause B.2.1 has to be supported.

14.8 Locators and content referencing

The table below lists the types of entity that may be addressed by locators in the present document, and defines the entities for which a text representation is required. In the case of locators, where a text representation is required the present document does not specify what that representation is, however terminal specifications based on GEM have to supply an unambiguous, concrete syntax for each of these entities.

NOTE: Clause 11.7.6 describes how this text representation can be used in DVB-J applications.

The present document does not require support for addressing any other type of entity in an MHP system by locator or URL.

Table 5: Addressable entities, locators and their text representation

Entity	Text Representation
Transport stream	locator text representation has to be defined
Network	no standardized text representation required
Service	locator text representation has to be defined
Service domain	locator text representation has to be defined.
Program Event	locator text representation has to be defined
MPEG Elementary Stream	locator text representation has to be defined
File	"file:", "http:" and "https:" URLs, as referred to in MHP [1], clause 14.8 locator text representation has to be defined for files located within a Service domain.
Directory	"file:", "http:" and "https:" URLs, as referred to in MHP [1], clause 14.8 locator text representation has to be defined for directories located within a Service domain.
Drip feed decoder	"dripfeed://"

14.9 Service identification

Java TV can support two kinds of locators for identifying a service: transport independent locators and transport dependent locators. Both enable global, unique identification of a service.

A transport independent locator has additional properties:

- It can identify two (or more) service instances as being the same service even if they for technical reasons have different transport dependent locators.

It is up to the service provider to decide whether different service instances are identified as being the same service.

- They can give alternate identifications for a single service.

Terminal specifications based on GEM shall provide a textual representation for transport dependent locators. Support for transport independent locators is not required by the present document.

14.10 CA system

The present document places no requirements on a CA system, if one is present. Thus, MHP [1], clause 14.10 is not included in the present document.

NOTE: GEM terminal specifications may, of course, include the MHP requirements in clause 14.10, by directly referencing the MHP specification as outlined in the present document in clause 4.1.3.2.

15 Detailed platform profile definitions

15.0 General

This clause defines the capabilities of platforms as presented to applications. Products that claim to conform to a profile shall provide at least the minimum capabilities identified for the profile. In some cases this implies that specific hardware resources are present in the platform.

Table 6: Platform profile definitions

Area	Specification	Enhanced Broadcast Profile 1	Interactive Broadcast Profile 1	Internet Access Profile 1
Static formats				
Bitmap pictures	MHP [1], clause 7.1.1.3, "PNG" + 15.1, "PNG - restrictions"	M	M	
	MHP [1], clause 7.1.1.3, "PNG" without restrictions	-	-	
	MHP [1], clause 7.1.1.4, "GIF"	-	-	
	MHP [1], clause 7.1.2 "MPEG-2 I-Frames"	M	M	
	MHP [1], clause 7.1.1.2 "JPEG" + 15.3, "JPEG - restrictions"	M	-	
	MHP [1], clause 7.1.1.2 "JPEG" without restrictions	-	M	
Audio clips	MHP [1], clause 7.1.4 "Monomedia format for audio clips"	M	M	
Video drips	MHP [1], clause 7.1.3 "MPEG-2 Video "drips""	M	M	
Text encoding	MHP [1], clause 7.1.5 "Monomedia format for text"	M	M	
Broadcast streaming formats				
Video	7.2.2, "Video"	M	M	
Audio	7.2.1, "Audio"	M	M	
Subtitles	7.2.3, "Subtitles"	-	-	
Fonts				
Built in	Character set see annex E Metrics see annex D Face: UK RNIB "Tiresias"	M	M	
Downloadable	7.4, "Downloadable fonts"	M	M	
Broadcast channel protocols				
	6.2.2, "MPEG-2 clauses"	M	M	
	6.2.5, "DSM-CC user-to-user object carousel"	-	-	
	IP Multicast stack based on: 6.2.6, "Protocol for delivery of IP multicast over the broadcast channel" 6.2.7, "Internet Protocol (IP)" 6.2.8, "User Datagram Protocol (UDP)"	O	Ro	
Interaction channel protocols				
TCP/IP	MHP [1], clause 6.3.3 "Transmission Control Protocol (TCP)" MHP [1], clause 6.3.2 "Internet Protocol (IP)"	-	M	
UDP/IP	6.2.8, "User Datagram Protocol (UDP)" MHP [1], clause 6.3.2 "Internet Protocol (IP)"	-	M	
HTTP	MHP [1], clause 6.3.7.1 "HTTP 1.1"	-	O	
DVB-J				
Core	11.3, "Fundamental DVB-J APIs"	M	M	
Presentation	MHP [1], clause 11.4.1, "Graphical User Interface API"	M (note MHP)	M (note MHP)	
	MHP [1], clause 11.4.2, "Streamed Media API"	M	M	
Data Access	MHP [1], clause 11.5.1, "Broadcast Transport Protocol Access API" as modified by 11.5, "Data access APIs"	M	M	
	MHP [1], clause 11.5.2, "Support for Multicast IP over the Broadcast Channel"	O	Ro	
	MHP [1], clause 11.5.3, "Support for IP over the Return Channel"	-	M	
	MHP [1], clause 11.5.4, "MPEG-2 clause Filter API"	M	M	
	MHP [1], clause 11.5.5, "Mid-Level communications API" as modified by 11.5, "Data access APIs"	-	M	
	MHP [1], clause 11.5.6, "Persistent Storage API"	M	M	
Service Information and Selection	11.6.1, "DVB service information API"	-	-	
	11.6.2, "Service selection API"	M	M	
	11.6.3, "Tuning API"	M	M	
	11.6.4, "Conditional access API"	-	-	
	11.6.5, "Protocol independent SI API"	M	M	
Common Infrastructure	11.7.1, "APIs to support DVB-J application lifecycle"	M	M	
	11.7.2, "Application discovery and launching APIs"	M	M	
	11.7.3, "Inter-application communication API"	M	M	
	11.7.4, "Basic MPEG concepts"	M	M	
	11.7.5, "Resource notification"	M	M	
	11.7.6, "Content referencing"	M	M	
	11.7.7, "Common error reporting"	M	M	

Area	Specification	Enhanced Broadcast Profile 1	Interactive Broadcast Profile 1	Internet Access Profile 1
Static formats				
Security	MHP [1], clause 11.8.1 "Basic Security"	M	M	
	MHP [1], clause 11.8.2 "APIs to Support TLS / SSL Over the Return Channel"	-	M	
	MHP [1], clause 11.8.3 "Additional permissions classes," except for the package org.dvb.net.ca.	M	M	
Others	11.9.1, "Timer support"	M	M	
	11.9.2, "User settings and preferences API"	M	M	
	11.9.3, "Profile and version properties"	M	M	
NOTE:	The javax.tv.graphics.TVContainer.getRootContainer method shall return an instance of org.havi.ui.HScene or null.			

Key	
-	Not applicable
O	Optional feature in the receiver
Ro	Recommended optional feature in the receiver
M	Mandatory feature in the receiver

Where a feature defined by MHP [1] is not included in the present document, it may be supported in a GEM terminal specification by referencing the relevant clause(s) of MHP [1] directly.

15.1 PNG - restrictions

MHP [1], clause 11.9.1 is included in the present document.

15.2 Minimum media formats supported by DVB-J APIs

MHP [1], clause 11.9.1 is included in the present document, with the following notes and modifications.

Support for subtitles is optional.

15.3 JPEG - restrictions

MHP [1], clause 15.3 is included in the present document.

15.4 Locale support

MHP [1], clause 15.3 is included in the present document.

NOTE: Terminal specifications may, of course, guarantee support for locales in addition to UK English, however, support for the UK English local is required by the present document.

15.5 Video raster format dependencies

This clause addresses the aspects of the present document that vary as a consequence of the video raster format.

15.5.1 Standard Definition (PAL/SECAM or NTSC resolution)

15.5.1.1 Logical pixel resolution

The logical pixel resolution shall be 72 dots per inch.

16 Registry of constants

16.1 System constants

MHP [1], clause 16.1 is included in the present document.

16.2 DVB-J constants

MHP [1], clause 16.2 is included in the present document, with the following notes and modifications.

Where this clause lists a constant for a class that is not required by the present document (e.g. a class in the package `org.dvb.si`), that constant is not required

Annex A (normative): External references; errata, clarifications and exemptions

MHP [1], annex A is included in the present document.

Annex B (normative): Broadcast filesystem and trigger transport

B.0 General

The present document does not specify a transport protocol for broadcast file systems or for trigger (event) delivery. It does, however, require that terminal specifications based on GEM provide a mechanism for delivery of filesystems and triggers.

NOTE: MHP [1], annex B defines a profile of DSMCC object carousels that fulfils the requirements of this annex.

B.1 Service domain

Terminal specifications based on GEM have to include a mechanism for signalling a service domain. A service domain is an entity that uniquely identifies a filesystem, which can contain files, directories, stream descriptions, trigger objects and trigger events. The format of these is described in the following clauses. This mechanism shall be sufficient to support all functionalities of the API defined in annex P.

Terminal specifications based on GEM shall define a syntax for a locator to refer to a service domain. This locator syntax shall support the encoding of an optional integer, in order to accommodate the requirements of the method `ServiceDomain.getLocator()` (see clause P.2.5.3).

A service domain provides a "mount point" for a file system. Once an application attaches to a service domain, a mapping from locators to files within the carousel is established. To read any object in a filesystem, a GEM application has to first mount a service domain, then navigate the filesystem to the desired object. The text form of a relative path to an object in the filesystem is represented by `dvb_abs_path` as specified in MHP [1], clause 14.1.

The signalling for a service domain shall be sufficient to identify the "root" directory of a filesystem, and allow attaching to that filesystem.

The details of mounting a service domain are described in annex P.

B.2 Filesystem requirements

B.2.1 Static requirements

Terminal specifications based on GEM have to include a mechanism for delivering a hierarchical file system within a service domain. It has to be possible to construct a locator that refers to files and directories in this hierarchy. The file system delivery mechanism has to satisfy the following minimum requirements. Of course, in addition to these limits, available bandwidth and memory resources would constrain the size of what can practically be broadcast.

Table B.1: Filesystem signalling requirements

Area	Minimum requirement
Characters Allowed in File and Directory Names	The ASCII character "a".."z", "A".."Z", "0".."9", "-", and "_". After the first character of a file name, "." and " " (the space character) is also permitted
Maximum length of file name	200 characters
Number of entries per directory	10 000
Maximum Directory Nesting	20 levels
Maximum File Length	$2^{27}-1$ bytes
Caching	See clause B.2.1.1

B.2.1.1 Caching behaviour

It shall be possible to signal along with carousel modules, information related to the caching behaviour of a GEM terminal.

This signalling shall contain sufficient information to derive the following:

Table B.2: Caching behaviour signalling requirements

Function	Type
priority_value	8 bit unsigned integer
transparency_level	8 bit unsigned integer

Semantics of these two fields are as defined in MHP [1], clauses B.2.2.4.2. and B.5.

GEM terminals that support caching shall comply with those semantics. Default behaviour is as defined in clause B.2.2.4.2.

B.2.2 Filesystem updates

It shall be possible to signal a new version of a file or directory.

B.3 Stream description

There shall be a signalling mechanism for sending a description of an MPEG stream. Such a stream can carry a service (e.g. in MHP, a DVB service).

NOTE 1: The following requirements are modelled on DSMCC BIOP::StreamMessage.

Stream descriptions shall be identified with a special file sent in the hierarchical filesystem described in clause B.2. This file shall contain information sufficient to derive the following:

Table B.3: Stream description

Function	Type
npt_source	Reference (see text)
stream_locator	Locator external form
duration	32 bit unsigned integer
audio_stream	flag
video_stream	flag
data_stream	flag
is_mpeg_program	flag

npt_source: A reference to a source of MPEG Normal Play Time (NPT). This shall be sufficient to derive NPT values, and the NPT rate. It may indicate that no source of NPT is associated with this stream collection.

stream_locator: A locator that references the streams of this collection.

duration: The duration of this stream description, in milliseconds. Signalling for this value is not required; if not present it shall always be considered to be zero.

NOTE 2: MHP signalling can indicate a value of up to 2^{32} seconds with a resolution of microseconds.

audio_stream: True if this stream contains audio.

video_stream: True if this stream contains video.

data_stream: True if this stream contains data.

is_mpeg_program: An indication whether or not this stream collection is an MPEG program.

B.4 Trigger signalling

B.4.0 General

There shall be a mechanism for signalling the presence of triggers to an application, and for delivery of those triggers.

B.4.1 Trigger object

Triggers shall be identified with a special file sent in the hierarchical filesystem described in clause B.2. This file shall contain information sufficient to derive all of the contents of a Stream description, plus the following:

Table B.4: Trigger object

Function	Type
num_triggers	16 bit unsigned integer
for (i=0; i<N; i++) { trigger_name event_id }	string 14 bit unsigned integer

num_triggers: The number of trigger events identified in this trigger object

trigger_name: The name of a trigger event. The signalling shall support trigger names up to 200 characters long containing any valid 7-bit ASCII character between 32 and 126, inclusive.

event_id: An integer uniquely identifying a trigger event within the context of the currently selected service.

NOTE: A trigger object corresponds to a BIOP::StreamEvent message in DSMCC.

B.4.2 Trigger event

It shall be possible to signal a trigger event. The signalling shall contain information sufficient to derive the following:

Table B.5: Trigger event

Function	Type
event_id	14 bit unsigned integer
is_do_it_now	flag
mpeg_npt	32 bit unsigned integer
payload	byte array

event_id: An integer uniquely identifying a trigger event within the context of the currently selected service.

is_do_it_now: Flag indicating if this is a "do it now" event. If true, this is a "do it now" event that is to be triggered upon reception. If false, this is a scheduled event to be triggered when a given NPT value is reached.

mpeg_npt: A normal play time value of an MPEG timebase. For "do it now" events, this value is ignored.

NOTE 1: A timebase associated with the stream identified by the Trigger object will be used by the terminal to send a trigger to a registered application.

payload: A sequence of up to 220 bytes containing arbitrary data.

NOTE 2: A trigger event corresponds to a DSMCC section carrying a stream event descriptor.

B.4.2.1 Extrapolation of NPT values

Terminal specifications based on GEM shall be written such that, for broadcasts confirming to appropriate broadcast norms and specifications and absent reception errors, any extrapolation of NPT values shall last no more than 5 seconds.

NOTE: This corresponds to the requirements in NPT signalling spelled out in MHP [1], clause B.2.4.4, "Timebases".

B.4.2.2 Monitoring of trigger events

Terminal specifications based on GEM shall require monitoring of at least one stream delivering scheduled stream events, and one stream delivering "do it now" stream events. For broadcasts confirming to appropriate broadcast norms and specifications and absent reception errors, the terminal shall raise an event in response to a scheduled trigger event provided that an application subscribed to the event at least 5 seconds before the scheduled time.

NOTE: This corresponds to the requirements spelled out in MHP [1], clause B.2.4.5, "Monitoring Stream Events." MHP requires that scheduled stream event descriptors be *broadcast* at least 5 s before the scheduled time, but in GEM this requirement is not expressed, because it is a part of appropriate broadcast norms and specifications.

Annex C (informative): References

C.1 Informative references from MHP

MHP [1], annex C is included in the present document.

C.2 Additional informative references

Reference	Edition	Description
[A]ARIB AE		Not published yet, please contact ARIB for more information. See http://www.arib.or.jp/ .

Annex D (normative): Text presentation

MHP [1], annex D is included in the present document, with the following notes and modifications.

Table D.3 in MHP [1], clause D.3.4.2 is not required for GEM terminal specifications that do not support a graphics device resolution of 720x576, however a functional equivalent has to be specified in GEM terminal specifications.

Annex E (normative): Character set

MHP [1], annex E is included in the present document.

Annex F (informative): Authoring and implementation guidelines

MHP [1], annex F is included in the present document.

Annex G (normative): Minimum platform capabilities

G.1 Graphics

MHP [1], clause G.1 is included in the present document, with the following notes and modifications.

The last bullet point of MHP [1], clause G.1.1 discusses a resolution of 720x576. This is to be interpreted as meaning the appropriate resolution with the terminal in standard definition mode.

MHP [1], clause G.1.5 defines minimum colour capabilities, including a precise definition of the CLUT. This CLUT may be inappropriate in some locations, because it is impossible to reproduce certain colours on all devices. For example, some of the specified colours fall outside of the NTSC gamut, and thus cannot be reproduced on devices with an NTSC composite video output. For this reason, the CLUT as specified in MHP is not required, but a functional equivalent has to be provided. It has to support at least 139 opaque colours, 48 colours at 30 % transparency, and 1 colour at 100 % transparency. It should provide a set of colours that approximates the MHP CLUT as closely as possible.

G.2 Audio

MHP [1], clause G.2 is included in the present document.

G.3 Video

MHP [1], clause G.3 is included in the present document.

G.4 Resident fonts and text rendering

MHP [1], clause G.4 is included in the present document, with the following notes and modifications.

Table G.2 in clause G.4.1 lists values for the number of TV lines that are not appropriate for all regions. GEM terminal specifications have to specify the functional equivalent of these values in order to produce an equivalent result in terms of a percentage of the screen size.

G.5 Input events

MHP [1], clause G.5 is included in the present document.

Some downloaded resident applications specified as extensions to the present document may perform some of the functions of the MHP navigator, e.g. the monitor application defined OCAP 1.0 [4]. In this case, such downloaded software has to implement a policy that is consistent with the requirements of the present document, e.g. MHP [1], clause G.5.

G.6 Memory

MHP [1], clause G.6 is included in the present document, with the following notes and modifications.

The bullet point requiring enough memory to load a 720x576 8 bit PNG image is to be read as requiring a PNG image whose size is the same as the resolution of an `HGraphicsDevice` at standard definition, as discussed in clause G.1.

G.7 Other resources

MHP [1], clause G.7 is included in the present document, with the following notes and modifications.

MHP [1], table G.4 refers to AIT clause filtering. As an MHP AIT is not required, this entry does not apply to the present document; however, attention is drawn to the requirement on detecting application description changes in clause 10.4.1.

The key lengths for application authentication in MHP [1], table G.4 apply only to codesigning using the MHP model.

Table G.5 in MHP [1], clause G.7 contains an entry for conditional access. The present document places no requirements on conditional access, thus this table entry is not included in the present document.

Annex H (normative): Extensions

MHP [1], annex H is included in the present document.

Annex I (normative): DVB-J fundamental classes

MHP [1], annex I is included in the present document.

Annex J (normative): DVB-J event API

MHP [1], annex J is included in the present document.

Annex K (normative): DVB-J persistent storage API

MHP [1], annex K is included in the present document.

Annex L (normative): User settings and preferences API

MHP [1], annex L is included in the present document, with the following notes and modifications.

In the class `org.dvb.user.GeneralPreference`, the preference "User Name" requires that name be reported as first name followed by last name. It is understood that "first name" and "last name" are ambiguous concepts in some locales. For this reason, in the present document this property is only required to contain the name of the user, in some order that is suitable for presentation to an end user.

Annex M:
Void

Annex N (normative): Streamed media API extensions

MHP [1] N is included in the present document, with the following notes and modifications.

References to "720x576" frames are to be read as referring to standard definition frames, as defined in the GEM terminal specification.

Annex O (normative): Integration of the JavaTV SI API

Terminal specifications based on GEM shall contain a mapping of the Java TV SI API to the network's underlying signalling. This mapping shall fulfil all of the semantic guarantees required by Java TV. The Java TV SI is formed by the classes in the package `javax.tv.si` and its subpackages, as defined for the present document, including any descriptive text accompanying those classes.

Annex P (normative): Broadcast transport protocol access

P.1 Overview

This clause includes a definition of APIs in the package `org.dvb.dsmcc`. The API is mapped to an *abstraction* of signalling based on DSMCC as specified in MHP [1]; no requirement to use DSMCC signalling is implied.

The API defined in this clause allows DVB-J applications direct access to information broadcast according to annex B. Of course, terminal specifications based on GEM may make other filesystems available via this API.

To benefit from the fact that most of the functionalities are already covered by the `java.io` package, this API inherits from `java.io` and only defines the extra functionalities pertaining to:

- a) the nature of a broadcast filesystem and its latency (e.g. possibility to asynchronously load the objects)
- b) the type of the objects that can be encapsulated in a carousel and that do not exist in a classical file structure. These are: `ServiceGateway`, `Stream` and `StreamEvent`.

An application can optionally use only the classes of `java.io`. Alternatively/additionally applications can use additional classes and methods adapted to the specific nature and latency of the network (such as for example, the asynchronous loading of objects).

The following, briefly explains the functionalities offered by this API.

The `ServiceDomain` class enables attaching to a Service domain.

When attached to a Service Domain, objects are available representing the types `File`, `Directory`, `Stream` description, `Trigger` object and `Trigger` event.

The class `DSMCCObject` is a common superclass for all of these types. It defines methods that deal with asynchronous or synchronous loading of Objects.

For the `File` and `Directory` Objects, their content is accessible as it would be for a classical file system, i.e. by using the `java.io` package (e.g. for listing the objects pointed to by a `Directory` object, you invoke the `list()` method of the `java.io.File` class, or to access the content of a `File`, you can instantiate a `FileInputStream` to read the `File`, etc.)

Additionally, the `DSMCCStream` and `DSMCCStreamEvent` classes define functionalities specific to the respective types of Objects (`Stream` description and `Trigger` object), enabling access to the attributes of these Objects. For the details of the attributes that can be accessed, refer to the documentation of these classes.

The `AsynchronousLoadingEvent` class and its subclasses represent events that are sent to a listener to notify it of the loading of an Object that had been activated by the application (asynchronous loading mode).

The `StreamEvent` class represents an abstraction of the real event that is generated, i.e. the `Trigger` event, which enables the broadcaster to synchronize the application with the stream. This class enables the access to the content of an event, as described in clause B.4.2.

Finally, the `StreamEventListener` and `AsynchronousLoadingEventListener` are interfaces that have to be implemented by the application, in order for it to receive the respective `StreamEvents` and `AsynchronousLoadingEvents`.

P.2 The org.dvb.dsmcc package

P.2.0 General

This package is derived from the `org.dvb.dsmcc` package as defined in MHP [1], annex P. A small number of the MHP methods are not required in terminal specifications based on GEM. Additionally, in GEM these methods are bound to the more abstract signalling requirements of annex B.

The description of each class from the `org.dvb.dsmcc` package defined in MHP [1], annex P is included in the present document, except as modified below.

In all cases, references to a "DSMCC object" in the signalling is to be interpreted as referring to the entity in the signalling represented by an object of type `DSMCCObject` in the present document.

P.2.1 DSMCCObject

The first sentence of the class description should be interpreted as meaning that this class represents objects in a Service domain.

P.2.2 DSMCCStream

References to `BIOP::Stream` message are to be interpreted as meaning the stream description as defined in clause B.3. References to the `BIOP::StreamEvent` message are to be interpreted as meaning the stream even description defined in clause B.4.1. References to elements of the `BIOP` messages are to be interpreted as referring to the corresponding element of the generic descriptions from annex B, as detailed below.

P.2.2.1 isAudio() method

This shall return true if the `audio_stream` flag indicates that this stream contains audio.

P.2.2.2 isData() method

This shall return true if the `data_stream` flag indicates that this stream contains data.

P.2.2.3 isMPEGProgram() method

This shall return true if the `is_mpeg_program` flag indicates that this object represents an MPEG program.

P.2.2.4 isVideo() method

This shall return true if the `video_stream` flag indicates that this stream contains video.

P.2.3 DSMCCStreamEvent

References to the `BIOP::StreamEvent` message are to be interpreted as meaning the stream even description defined in clause B.4.1. References to elements of the `BIOP` messages are to be interpreted as referring to the corresponding element of the generic descriptions from annex B, as detailed below.

Throughout this class, references to a `DSMCC StreamEvent` in the signalling are to be read as referring to a trigger object, as defined in clause B.4.1.

P.2.4 InvalidFormatException

This exception may be thrown when any inconsistency in the underlying signalling is received.

P.2.5 ServiceDomain

The first paragraph of the class description is to be replaced by the following: A ServiceDomain represents the entity described in clause B.1.

Throughout this class, references to "service gateway" or "service domain" are to be interpreted as referring to service domain, as described in clause B.1.

P.2.5.1 ServiceDomain.attach(byte[])

ServiceDomain.attach(byte[]) is required by the present document. The syntax and usage of the NSAP address is out of the scope of the present document.

P.2.5.2 ServiceDomain.attach(Locator) and attach(Locator, int)

The locator parameter is to be interpreted as any locator that refers to a service domain. Locator formats are discussed in clause 14.8.

P.2.5.3 ServiceDomain.getLocator()

The description of this method is considered to read as follows:

Return the locator for this service domain. If this ServiceDomain instance was last attached by specifying a locator then that exact same locator shall be returned. If the attach was done with the attach(locator, int) signature, the locator is complemented with a representation of the integer.

P.2.5.4 ServiceDomain.getNSAPAddress()

Signalling to support the ServiceDomain.getNSAPAddress() method is not required by the present document. In terminal specifications where no such signalling is defined, the behaviour of invoking this method may be undefined.

P.2.5.5 ServiceDomain.getURL(Locator)

The description of this static method is considered to read as follows:

Returns a URL corresponding to a locator referring to a file or a directory, as specified in table 5. If the service domain corresponding to the locator is attached and the file or directory referenced in the locator exists then an instance of java.net.URL is returned which can be used to reference this object.

Parameters:

l - a locator referring to a file or directory, as specified in table 5.

Returns:

a java.net.URL which can be used to access the file or directory referenced by the locator.

Throws:

InvalidLocatorException - if the locator is not a valid locator or does not includes all elements leading to a file or directory.

NotLoadedException - is thrown if the locator is valid and includes enough information but it references a service domain which is not attached.

FileNotFoundException - if the service domain is attached but the file or directory referenced by the locator does not exist.

P.2.6 ServiceXFRErrorEvent

This class is required by the present document, however signalling that would cause this error to be generated is not required by the present document.

P.2.7 ServiceXFRException

This class is required by the present document, however signalling that would cause this exception to be generated is not required by the present document.

P.2.8 ServiceXFRReference

This class is required by the present document, however signalling that would cause an event containing an object of this type is not required by the present document.

P.2.9 StreamEvent

Throughout this class, references to the DSMCC stream event descriptor are to be read as referring to the trigger event, as described in clause B.4.2. References to the event data refer to the payload defined in that clause.

Annex Q (normative): Datagram socket buffer control

MHP [1], annex Q is included in the present document.

Annex R (normative): DVB-J return channel connection management API

MHP [1], annex R is included in the present document.

Annex S (normative): Application listing and launching

MHP [1], annex L is included in the present document.

Annex T (normative): Permissions

MHP [1], annex T is included in the present document.

Annex U (normative): Extended graphics APIs

MHP [1], annex U is included in the present document.

Annex V:
Void

Annex W (informative): DVB-J examples

W.1 DVB-J examples from MHP

MHP [1], annex W is included in the present document.

W.2 Example of enumeration extension

To illustrate the importance of the requirement in clause 4.1.4.1, consider an application that is written to the GEM specification which wishes to query the type of return channel connection, and react accordingly. Such code might be written in the following manner:

```
import org.dvb.net.rc.RCInterface;
import org.dvb.net.rc.RCInterfaceManager;

public class AppRCTester {

    /**
     * Set up the return channel.
     * @return true if it was successfully set up, false otherwise.
     */
    public boolean setUpRC() {
        RCInterface[] ifs = RCInterfaceManager.getInstance().getInterfaces();
        boolean success = false;
        for (int i = 0; !success && i < ifs.length; i++) {
            RCInterface inter = ifs[i];
            switch(inter.getType()) {
                case TYPE_CATV:
                    success = setupCATV(inter);
                    break;
                case TYPE_DECT:
                    success = setupDECT(inter);
                    break;
                case TYPE_ISDN:
                    success = setupISDN(inter);
                    break;
                case TYPE_LMDS:
                    success = setupLMDS(inter);
                    break;
                case TYPE_MATV:
                    success = setupMATV(inter);
                    break;
                default:
                    // Do nothing - this always fails
            }
            return success;
        }
        .... definition of methods setupCATV et al.
    }
}
```

If it were permissible for a GEM terminal specification to sub-divide TYPE_CATV by introducing new values into the enumeration, then this code would always fail.

If a terminal specification needs to sub-divide the values of an enumeration, it may do so by introducing a new method to report the sub-divisions. For example, to sub-divide `TYPE_CATV`, a terminal specification could introduce an interface and a set of values like the following:

```
package org.specbody.net.rc;

/**
 * On specbody terminals, all instances of org.dvb.net.rc.RCInterface
 * for which getType() returns TYPE_CATV shall implement this interface.
 */

public interface CATVRCInterface {

    public final static int TYPE_CATV_SUBTYPE_1 = 1;
    public final static int TYPE_CATV_SUBTYPE_2 = 2;
    public final static int TYPE_CATV_SUBTYPE_3 = 3;

    /**
     * @returns one of TYPE_CATV_SUBTYPE_1, TYPE_CATV_SUBTYPE_2
     *             or TYPE_CATV_SUBTYPE_3
     */
    public int getCATVType();

}
```

Note that this extension mechanism works for `org.dvb.net.rc.RCInterface` because instances of this class are always created by a factory method that is a part of the platform. This particular method for extending the behaviour of GEM would not work if the enumeration value were returned by a method in a class with a constructor that is accessible to applications, because it would be impossible to mandate that all instances conforming to certain criteria implement an additional interface. In this case, other extension mechanisms would need to be employed.

Annex X (normative): Test support

MHP [1], annex X is included in the present document.

Annex Y (normative): Inter-application and Inter-Xlet communication API

MHP [1], annex Y is included in the present document.

Annex Z (informative): Services, service contexts and applications in an MHP environment

MHP [1], annex Z is included in the present document, with the following notes and modifications.

This informative clause includes references to some signalling details not required by the present document. Where this is the case, it is to be read as an example of one possible way of fulfilling the abstract requirements placed on terminal signalling by the present document.

History

Document history		
V1.1.1	January 2003	Publication