ETSI TS 102 819 V1.4.1 (2008-05)

Technical Specification

Digital Video Broadcasting (DVB); Globally Executable MHP version 1.0.3 (GEM 1.0.3)

Reference
RTS/JTC-DVB-220

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, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

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 2008.
© European Broadcasting Union 2008.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™, **TIPHON**™, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered 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

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).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

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 [3], and ARIB AE [5]. 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 GEM terminal specifications, 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

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.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

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

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] ETSI ES 201 812 (V1.1.2): "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.0.3".

[2] ETSI TS 101 154 (V1.8.1): "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream".

[3] OCAP 1.0: "OpenCable Application Platform version 1.0".

NOTE: Available at http://www.opencable.com/specifications/.

[4] ISO 639 (all parts): "Codes for the representation of names of languages".

[5] ARIB STD-B23 (V1.1) (February 2004): "Application Execution Engine Platform for Digital Broadcasting".

NOTE: Available at http://www.arib.or.jp.

[6] IETF RFC 3073: "Portable Font Resource (PFR) - application/font-tdpfr MIME Sub-type Registration".

[7] tm2971r2: "MHP Specification Version 1.0.3 Errata #3".

NOTE: Available at http://www.mhp.org/mhp_technology/mhp_1_0/tm2971r2-tam801r11-mhp-103-errata3-clean.pdf.

[8] ANSI/SCTE 90-1 2005: "SCTE Applications Platform Part 1 OCAP 1.0 Profile".

NOTE: Available at http://www.scte.org/content/index.cfm?pID=1217.

[9] System Description Blu-ray Disc Read-Only Format.

NOTE: http://www.blu-raydisc.info/format_spec.php

[10] ETSI EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

[11] ISO/IEC 10646: "Information technology - Universal Multiple-Octet Coded Character Set (UCS)".

[12] JSR927: Java TVTM API 1.1.

NOTE: The latest release of JavaTV is at http://www.jcp.org/en/jsr/detail?id=927. JavaTV 1.1.1 is the current maintenance release of the JavaTV 1.1 API, with no semantic changes against JavaTV 1.1.

[13] DAVIC 1.4.1 Specification Part 9 (June 1999): "Complete DAVIC Specifications".

NOTE: Available at http://www.davic.org.

2.2 Informative references

[14] ETSI TS 102 812: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.1".

[15] W3C Recommendation 20060816: "XML Namespaces".

NOTE: Available at http://www.w3.org/TR/REC-xml-names.

[16] Application Definition Blu-ray Disc Format - BD-J Baseline Application and Logical Model Definition for BD-ROM, March 2005.

NOTE: Available at http://www.blu-raydisc.com/ under "Technical Info", "Public Specifications".

[17] ETSI TR 101 162: "Digital Video Broadcasting (DVB); Allocation of Service Information (SI) codes for DVB systems".

[18] ETSI ETS 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

NOTE: Available at http://www.dvb.org.

2.3 Superseding references

GEM terminal specifications are allowed to supersede references used in MHP. For the features used by the GEM terminal specification, subsequent versions of those referenced specifications must provide features that are fully backwards compatible with the version used by MHP.

NOTE: It is the responsibility of the organization writing the GEM terminal specification that superseding references are compatible with MHP.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions given in TS 101 812 [1] and the following apply:

3.1.1 Definitions from MHP

MHP [1], clause 3.1 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:

Competitive API: API defined in a GEM terminal specification that is used to access functionality (e.g. signalling) that could be reasonably mapped to a GEM API

NOTE: A competitive API would be used to access functionality that is functionally equivalent to a GEM requirement. Competitive APIs are forbidden, unless the functionality is also exposed via the GEM API. See clause 4.1.4.2, "Competitive APIs".

Complimentary functional equivalent: functionality (e.g. signalling) in a GEM terminal specification that is functionally equivalent to a GEM functional equivalent, but where another mechanism already exists that satisfies the GEM requirement that a functional equivalent be defined

NOTE: In this case, the two functionally equivalent mechanisms are called complementary functional equivalents

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.

enumeration: type that includes in its definition an exhaustive list of possible values for variables of that type

NOTE: In Java, enumerations are not directly supported, but they are often simulated with a set of integer constants.

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 the present document that do not require literal conformance with the corresponding requirement in the underlying MHP specification, but allow for a compatible substitution.

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.

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 TS 101 154 [2]

target: category of GEM terminal specification(s), determined by the mechanism used to transport applications from the content producer to the viewer

EXAMPLE: The broadcast target is for use by GEM terminal specifications in broadcast environments; the packaged media target is used for GEM terminal specifications where the media is packaged onto a physical carrier which is possibly read-only, such as an optical disc.

trigger: event that may cause a change in the behaviour of an application that registers interest in such events

NOTE: Triggers come from the broadcast stream. The trigger may include a reference to time relative to the NPT of a media stream or be asynchronous. It also can carry some semantically significant payload in order to affect changes in an application based on information not available at the time an application was written.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given 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
EPG	Electronic Programme Guide
GEM	Globally Executable MHP
ID	IDentifier
IP	Internet Protocol
MHP	Multimedia Home Platform
MPEG	Moving Picture Experts Group
NPT	Normal Play Time
NTSC	National Television Systems Committer
OCAP	OpenCable Applications Platform
POD	Point Of Deployment
SI	Service Information
TCP	Transmission Control Protocol
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 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 shall 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 as described in clause 15.6, "Functional Equivalents".

4.1.4 Addition of non-GEM interfaces

GEM terminal specifications may add public interfaces, provided that they are added in a namespace that does not conflict with GEM. For example, OCAP 1.0 [3] 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 clause W.2, "Example of enumeration extension".

4.1.4.2 Competitive APIs

A GEM terminal specification may include functionality that is functionally equivalent to functionality required by GEM, and thus could reasonably be mapped to a GEM API. This functionality (e.g. signalling) might be in addition to functionality that satisfies the GEM requirement that a functional equivalent be defined. In other words, the two mechanisms may be complementary functional equivalents.

In this case, the GEM terminal specification shall define a mapping to the GEM API for both mechanisms, that is, for both functional equivalents. All such mappings shall comply with the GEM requirements for the functional equivalents.

As with all functional equivalents, complementary functional equivalents may include features beyond what is required by GEM, and that can not reasonably be exposed via a GEM API. In this case, it is of course allowable for the GEM terminal specification to define additional API to expose these features, in a non-GEM package namespace.

NOTE: Consider, for example, the "Carousel" functional equivalent. A GEM terminal specification might provide two different signalling schemes for delivering the files and associated events for a GEM application. If this is done, GEM-compliant bindings will be defined for the APIs and other requirements of GEM for both signalling schemes. This is illustrated in the figures 1 and 2.

4.1.4.2.1 Illustration of Complementary Functional Equivalents

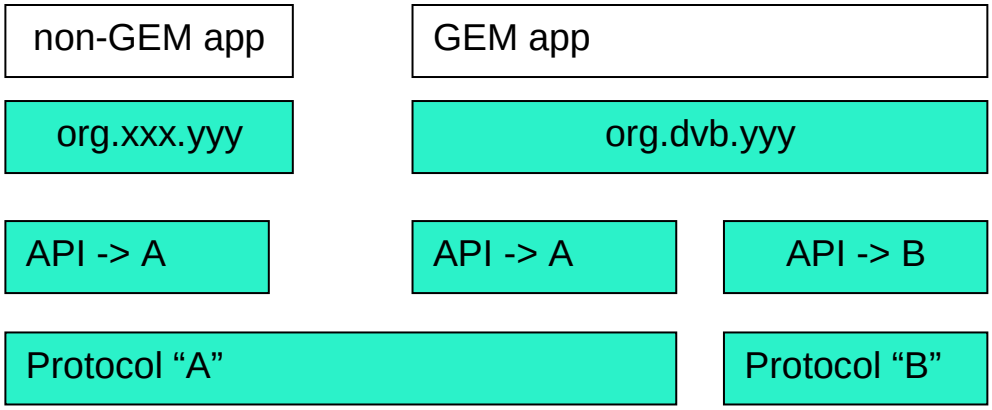

Figure 1: Complementary Functional Equivalents

Figure 1, showing complementary functional equivalents A and B, both mapped to the GEM API org.dvb.yyy. Mechanism A includes additional features, which are exposed to non-GEM applications via the API org.xxx.yyy. This situation is normal and does not conflict with GEM clause 4.1.4.2.

4.1.4.2.2 Illustration of Competitive APIs

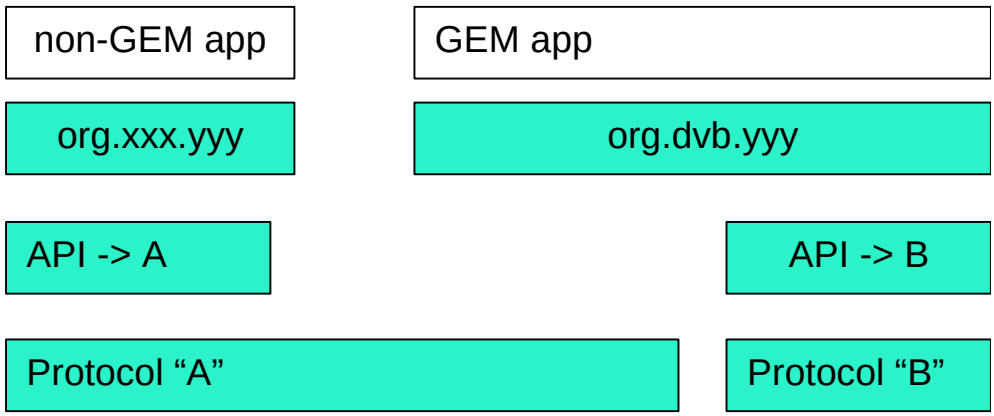

Figure 2: Competitive APIs

Figure 2, showing complementary functional equivalents A and B. Only mechanism B is mapped to the GEM API; A is only exposed via the non-GEM API org.xxx.yyy. In this case, org.xxx.yyy is a competitive API, and is in conflict with clause 4.1.4.2, "Competitive APIs".

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 conformance with the present document

To be fully conformant with the present document, GEM terminals shall conform to at least one of the profiles defined in clause 15, "Detailed platform profile definitions". Further, GEM terminals shall be conformant with any one of the following specifications:

- TS 101 812 [1] (MHP 1.0).
- OCAP 1.0 [3].
- SCTE 90-1 [8].
- Blu-Ray Disc Player Specification [9].

NOTE 1: All realizations of GEM functional equivalents and other information about the Blu-Ray Disc Player Specification, of particular interest to application authors, is available in the BD-J Baseline Application and Logical Model Definition for BD-ROM [16].

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

GEM terminal specifications shall contain a normative requirement that its terminals conform with GEM. GEM terminal specifications shall further require that, in the event of a conflict between GEM and the GEM terminal specification, GEM shall take precedence. Any errata to GEM shall be in a designated section of GEM errata, which shall only contain errata agreed for publication in a subsequent version of GEM. The text of the errata may be in the GEM terminal specification, but there shall be a statement referring to a document under the control of the DVB that confirms that these are agreed errata.

NOTE 2: As an example, this can be implemented with language like the following, drafted for a fictional GEM terminal specification called "A GEM Terminal Specification" (AGTS).

7.1 Compliance with GEM

AGTS terminals shall comply in full with GEM [1]. The present document adopts the MHP definition of the following functional equivalents, as specified in GEM [1] clause 15.6:

- Arch
- Carousel
- Text Wrapping

For avoidance of doubt, in the event of a conflict between GEM [1] and the present document, the normative guarantees of GEM [1] shall take precedence except as detailed in clause 7.1.1, "GEM errata".

7.1.1 GEM errata

Following are errata to GEM [1]. The changes presented have been agreed by the appropriate DVB subgroup for publication in a subsequent version of GEM [1].

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 shall be interpreted as referring to the appropriate clause of GEM, unless there is an explicit statement to the contrary in the present document.

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 GEM terminal specifications. 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 GEM terminal specifications, 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 [3] 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 may carry GEM applications.

4.2.3 Inclusion of clauses of the MHP specification

Unless otherwise noted, inclusion of a clause or annex of MHP [1] implies inclusion of all sub clauses.

4.2.4 Conventions within the present document

Use of the term "GEM" within a normative clause of the present document shall be interpreted as referring to the present document.

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 GEM terminal specifications.

6 Transport protocols

6.1 Introduction

For broadcast targets, 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.

NOTE: MHP [1] clause 6.1 provides background material on the transport protocols for MHP terminals.

For packaged media targets, the GEM terminal can navigate AV streams on the physical carrier and communicate with the external world through different IP-based networks. Wherever GEM refers to broadcast-related services or protocols, a packaged media target uses streams and formats from the packaged media. Note that the MHP/GEM term "interaction channel" refers to IP-based connectivity.

6.2 Broadcast channel protocols

For broadcast targets, 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. See clause 15.6, "Functional Equivalents".

The full details of APIs that provide DVB-J applications with access to broadcast protocols are in clause 9.The last paragraph of MHP [1], clause 6.2 places a normative requirement related to conditional access descrambling and the section filter API. This requirement applies to GEM terminal specifications that include the MHP definition of the functional equivalent named "Conditional Access" as specified in clause 15.6, "Functional Equivalents".

NOTE 2: For the packaged media profiles, these definitions might not be used.

6.2.1 MPEG-2 transport stream

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

6.2.2 MPEG-2 sections

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 Object carousel

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

Use of the Object Carousel protocol as defined by MHP [1], clause 6.2.5 is not required for GEM terminal specifications. MHP [1], clause 6.2.5 shall apply to GEM terminal specifications that include the MHP definition of the functional equivalent named "Object Carousel" as specified in clause 15.6, "Functional Equivalents". GEM terminal specifications that do not include this MHP definition shall specify a functional equivalent that satisfies the requirements of the API specified in Annex P, MHP [1], clause 6.2.5 is an example of such a mechanism, and is informative for these GEM terminal specifications.

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

MHP [1], clause 6.2.5.2 is not relevant to the present document.

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 GEM terminal specifications that do not include the MHP definition of the functional equivalent named "Object Carousel" in 15.6, "Functional Equivalents", therefore the reference to MHP [1], clause B.2.11 does not apply to these GEM terminal specifications. However, GEM terminal specifications 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 to all GEM terminal specifications. Thus, the language in MHP [1], clause 6.2.5.3 following the first paragraph do 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 the DVB Multiprotocol Encapsulation protocol is not required for GEM terminal specifications. If, however clause 11.5.2, "Support for Multicast IP over the Broadcast Channel" is supported, some mechanism for delivery and signalling of IP multicast over the broadcast channel shall be specified.

NOTE: 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 Service information

MHP [1], clause 6.2.9 is included in the present document, with the following notes and modifications.

Use of DVB service information is not required for GEM terminal specifications, however some mechanism for delivery of service information that is sufficient for the SI access mechanisms required by GEM shall be specified. MHP [1], clause 6.2.9 may be taken as an informative example of such a mechanism for GEM terminal specification that do not include the MHP definition of the functional equivalent named "SI" as specified in clause 15.6, "Functional Equivalents".

6.2.10 IP signalling

MHP [1], clause 6.2.10 is included in the present document, with the following notes and modifications.

Use of this signalling is not required for GEM terminal specifications, however a functional equivalent that satisfies the requirements of MHP [1], clause 11.5.2 is required if support for IP over the broadcast channel is supported.

NOTE: This feature is optional in all profiles of GEM.

6.3 Interaction channel protocols

MHP [1], clause 6.3 is included in the present document, with the following notes and modifications. For this clause, subclauses are only included as explicitly indicated.

Some of the protocols shown in the diagram are not required by the present document in any profile.

The diagram is replaced with the following one.

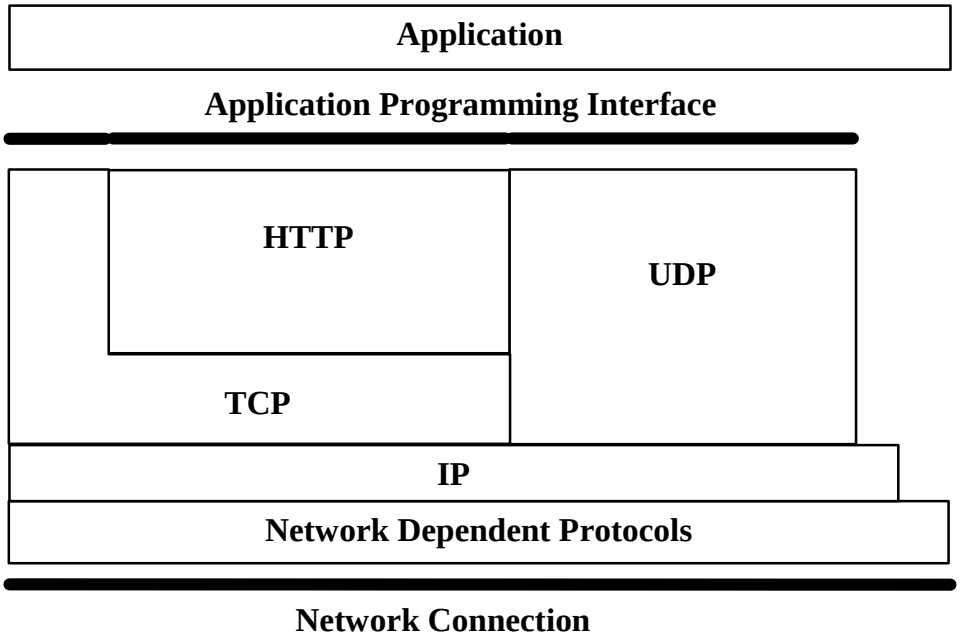

Figure 3: Interaction Channel Protocol Stack

6.3.1 Network Dependent Protocols

MHP [1], clause 6.3.1 is included in the present document, with the following notes and modifications.

NOTE: The protocols listed in this clause are not explicitly required in any profile of the present document, or of any profile in MHP [1].

6.3.2 Internet Protocol

MHP [1], clause 6.3.2 is included in the present document.

6.3.3 Transmission Control Protocol

MHP [1], clause 6.3.3 is included in the present document.

6.3.4 UNO-RPC

MHP [1], clause 6.3.4 is included in the present document.

6.3.5 UNO-CDR

MHP [1], clause 6.3.5 is included in the present document.

6.3.6 DSM-CC User to User

MHP [1], clause 6.3.6 is included in the present document.

6.3.7 Hypertext Transfer Protocol (HTTP)

6.3.7.1 HTTP 1.1

MHP [1], clause 6.3.7.1 is included in the present document.

NOTE: HTTP 1.1 support as specified is not required in any profile or target of GEM. This does not preclude a GEM terminal specification using a different profile of HTTP.

6.3.8 Service Specific

MHP [1], clause 6.3.8 is included in the present document, with the following notes and modifications.

GEM terminal specifications that wish to include the functionality provided by this definition may substitute a functional equivalent.

NOTE: This functionality is not required by any profile in the present document.

6.3.9 User Datagram Protocol

MHP [1], clause 6.3.9 is included in the present document.

6.3.10 DNS

MHP [1], clause 6.3.10 is included in the present document.

7 Content formats

NOTE: This clause contains definitions referenced from other parts of the present document. Use of these formats may be optional, or it may be possible to replace them with a functional equivalent.

7.1 Static formats

7.1.1 Bitmap image formats

7.1.1.1 Image encoding restrictions

MHP [1], clause 7.1.1.1 is included in the present document.

7.1.1.2 JPEG

MHP [1], clause 7.1.1.2 is included in the present document.

NOTE: Other possible restrictions on JPEG are described in clause 15.3, "JPEG - restrictions".

7.1.1.3 PNG

MHP [1], clause 7.1.1.3 is included in the present document, with the following notes and modifications:

GEM terminal specifications may allow or require processing of colour space or gamma information in image transformations.

7.1.1.4 GIF

MHP [1], clause 7.1.1.4 is included in the present document.

7.1.2 MPEG-2 I-Frames

MHP [1], clause 7.1.2 is included in the present document.

7.1.3 MPEG-2 Video "drips"

MHP [1], clause 7.1.3 is included in the present document.

7.1.4 Monomedia format for audio clips

MHP [1], clause 7.1.4 is included in the present document, with the following notes and modifications.

Use of the MPEG-1 format as defined by MHP [1], clause 7.1.4 is not required for GEM terminal specifications for packaged media targets. MHP [1], clause 7.1.4 shall apply to GEM terminal specifications that include the MHP definition of the functional equivalent named "Audio Clips" as specified in clause 15.6, "Functional Equivalents". GEM terminal specifications that do not include this MHP definition shall specify a functional equivalent that satisfies the requirements of the APIs mentioned in the MHP table as included in clause 15.2, "Minimum media formats supported by DVB-J APIs" of the present document. MHP [1], clause 7.1.4 is an example of such a media type, and is informative for these GEM terminal specifications.

7.1.5 Monomedia format for text

MHP [1], clause 7.1.5 is included in the present document.

7.2 Broadcast streaming formats

For packaged media targets, the term "broadcast streaming formats" should be interpreted as "stream formats".

7.2.1 Audio

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

7.2.2 Video

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

7.2.3 Subtitles

MHP [1], clause 7.2.3 is included in the present document, with the following notes and modifications.

Support for DVB subtitles, teletext subtitles or any other form of subtitles is optional in the present document.

NOTE: OCAP 1.0 [3] 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, with the following notes and modifications.

The inclusion of resident fonts on a GEM terminal as specified by MHP [1], clause 7.3 is not required for GEM terminal specifications for packaged media targets. MHP [1], clause 7.3 shall apply to GEM terminal specifications that include the MHP definition of the functional equivalent named "Resident Fonts" as specified in clause 15.6, "Functional Equivalents".

NOTE: `java.awt` requires that a default font be available, but does not specify what it is. It is recommended that applications explicitly specify a known font for consistent results.

7.4 Downloadable fonts

MHP [1], clause 7.4 is included in the present document, with the following notes and modifications.

Use of the PFR downloadable font format as defined by MHP [1], clause 7.4 is not required for GEM terminal specifications for packaged media targets. MHP [1], clause 7.4 shall apply to GEM terminal specifications that include the MHP definition of the functional equivalent named "Downloadable Fonts" as specified in clause 15.6, "Functional Equivalents". GEM terminal specifications that do not include this MHP definition shall specify a functional equivalent that enables a font to be packaged with GEM applications. MHP [1] clause 7.4 is an example of such a format, and is informative for these GEM terminal specifications.

7.5 Colour representation

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

7.6 MIME types

MHP [1], clause 7.6 is included in the present document, with the following notes and modifications:

GEM terminal specifications may replace the entry "application/dvb.pfr" with "application/font-tdpfr" as defined in RFC 3073 [6] if they wish.

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 of the present document. 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 of the present document. 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.

NOTE: It is not specified whether or not reselecting the current service will restart applications signalled with a true value for the `service_bound_flag` defined in clause 10.4.3.

GEM terminal specifications for packaged media targets may not assume applications offering EPG functionality. In such a case, the language in the first paragraph of clause 9.1.1 in MHP [1] relating to applications offering such EPG functionality is to be more generally interpreted as applications offering service selection functionality.

For packaged media targets, a service is typically not the contents of a broadcast service, but instead, the contents is stored on a persistent storage medium. Therefore, for packaged media targets, the sentence in MHP [1] clause 9.1.1:

> "In this version of the specification, the service is the contents of a broadcast DVB service, including audio/video streams, data streams and all the service information, applications and application signalling that is being broadcast."

is extended with the following:

> "For packaged media targets, a service is made up of elements that are typically stored on a storage medium, including audio/video streams, data streams, service information, applications and application signalling."

GEM terminal specifications based on the packaged media profile may not support a tuner. In such a case, MHP [1], clause 9.1.7 does not apply.

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 of the present document.

GEM terminal specifications based on the packaged media profile may not support the dynamic update of a service's AIT. In such a case, corresponding language in MHP [1] clause 9 and its subclauses does not apply.

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 of 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 [3]. In this case, such downloaded software shall 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 shall be used or the packaging of applications; this is left for GEM-based specifications to define.

MHP [1], clause 10.1 and its subclauses contains introductory remarks about MHP's signalling and may be taken as informative.

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 (see clause 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 (see clause 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 shall contain information sufficient to locate the following:

- the Application Description (see clause 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 shall 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 shall 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 and Tuning

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.

GEM terminal specifications for packaged media targets may not support a tuner. In such a case, the above paragraph does not apply.

10.4.3 Content of the application description

The Application Description describes applications and their associated information. It shall 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 GEM terminal specifications.

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 last 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 shall 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 shall 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 [4] language code with each application name. The GEM terminal specification shall define which text encodings are required to be supported for this name.

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.

NOTE 1: MHP [1] defines a test_application_flag that allow test applications to be broadcast without being displayed on consumer equipment that has not been put in a special "test mode". Equivalent functionality is *not* required by GEM, however, GEM terminal specification authors may wish to consider such functionality.

NOTE 2: GEM terminal specifications may define application icon signalling that also allows different icon sizes to be indicated.

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 shall be present in GEM terminal specifications.

10.5.2 Content of DVB-J application description

For each application description that refers to a DVB-J application, it shall 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 shall implement the Xlet interface. The signalling shall 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.

10.6 Constant Values

GEM terminal specifications that include the MHP definition of the functional equivalent named "Application Signalling" as defined in clause 15.6, "Functional Equivalents" and that define additional descriptors in the AIT shall register the descriptor tag values with the DVB. For informative purposes current registrations are shown in table 3.

Table 3

AIT Descriptor Tag Values	Organization
0x060 to 0x065	**ATSC**
0x066 to 0x069	**Available for registration**
0x06A to 0x06D	**Cable Labs**
0x06E to 0x07F	**Available for registration**

These values are formally maintained in TR 101 162 [17].

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, with the following notes and modifications.

MHP [1], clause 11.2.11.1 applies only for GEM terminal specifications that include the MHP definition of the functional equivalent named "SI" as defined in clause 15.6, "Functional Equivalents".

MHP [1] clause 11.2.1 requires that "MHP terminals shall ensure that any error conditions due to inability to link classes in DVB-J applications are thrown only if the execution of the application reaches a bytecode where these classes are referenced from the application. The presence of the reference in the class file to an non-existing class shall not lead to an error being thrown when loading the referencing class". This requirement does not apply to GEM.

NOTE: Removing this requirement enables optimized "eager linking" VM implementations. Applications that need to test for features that are optionally present may use a standard Java idiom for this. One such idiom is documented in clause W.3.

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: As with other sub clauses, MHP [1], clause 11.3.1.1 bullet point g applies to the present document. Thus, all GEM terminal specifications 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. For GEM terminal specifications that do not include the MHP definition of the functional equivalent named "Carousel" as defined in clause 15.6, "Functional Equivalents," this requirement does not apply; however, if the functionally 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.

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 6.3.3.

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 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 as described in clause 15.6, "Functional Equivalents".

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, GEM terminal specifications should 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.

NOTE 3: Alters the definition of the constant `java.awt.event.KeyEvent.CHAR_UNDEFINED`. The return value of `java.awt.event.KeyEvent.getKeyChar()` is defined in terms of `CHAR_UNDEFINED`. This value may only be returned for `KEY_PRESSED` and `KEY_RELEASED` events; it is never returned for `KEY_TYPED` events. In no case should an interoperable application rely on the return value of `getKeyChar()` for a `KEY_PRESSED` or `KEY_RELEASED` event; further, the value returned for an undefined key is not defined interoperably and may vary across platform implementations.

The reference to "platform resident fonts" in MHP [1], clause 11.4.1.1 is to be interpreted as "resident fonts", as described in clause 7.3, "Resident fonts". These fonts are available e.g. to the AWT APIs without using the downloadable font API specified in Annex U.

The text in MHP [1] clause 11.4.2.3 item b defines a default for service component presentation. GEM terminal specifications may define additional mechanisms that overrides this default.

The last paragraph of MHP [1], clause 11.4.2.2 places a requirement on the handling of `DVBLocators`. For GEM terminal specifications that do not include the MHP definition of the functional equivalent named "Content Referencing" as defined in clause 15.6, "Functional Equivalents", 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 6.3.3.

The following classes are required by MHP [1], clause 11.4.2.5.1 but are not required by the present document:

- `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.`

The following class is required by MHP [1], clause 11.4.2.5.2 but is not required by the present document:

- `org.davic.media.SubtitlingLanguageControl.`

For the packaged media target, the following classes are required by MHP [1], clause 11.4.2.5.2 but are not required by the present document:

- `org.davic.media.FreezeControl`
- `org.davic.media.MediaFreezeException`

Some of the classes referenced in MHP [1], clauses 11.4.2.5.4 and 11.4.2.5.5 are not required as specified by other clauses in the present document.

The title of the clause 11.4.2.5.4, "Required controls for broadcast profiles" is considered to read as "Required controls for broadcast profiles and packaged media profiles".

The following class is required by MHP [1], clause 11.4.2.5.2 but is only required by the present document for terminal specifications that do include the MHP conditional access API:

- `org.davic.media.NotAuthorizedMediaException`

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. This clause also places semantic requirements on subtitling-related APIs. As these APIs are optional in GEM, these requirements apply only if these APIs are required by the GEM terminal specification.

11.5 Data access APIs

11.5.1 Broadcast Transport Protocol Access API

MHP [1], clause 11.5.1 is included in the present document, with the following notes and modifications.

The reference to annex P is to be read as referring to annex P of the present document.

Relative file names used to access objects in the application filesystem shall be taken as being relative to the base directory indicated in the application description defined in clause 10.5. This is the GEM abstraction of MHP's application location descriptor, referenced in paragraph two of MHP [1] clause 11.5.1.

NOTE 1: Absolute and relative paths may be used in accordance with the APIs concerned, e.g. with `java.net.URL` (when used with a `"file://"` string prefix) and with `java.io.File`.

The reference to clause B.5 is to be read as referring to clause B.2.1.1 of the present document.

In clause 11.5.1.1, the fifth bullet point, which addresses the method `lastModified()` is replaced by:

- The method `lastModified()` returns a value as specified in the GEM terminal specification. GEM terminal specifications shall define signalling for this value to indicate at least 256 distinct values.

NOTE 2: Even if the MHP definition for the "carousel" functional equivalent is adopted by a GEM terminal specification, the above applies. Such specifications need to define the binding for `lastModified()`, even if this binding is the same as MHP's.

NOTE 3: GEM terminal specifications may, of course, define a binding that provides for a larger set of values than the minimum required for the method `lastModified()`.

11.5.2 Support for Multicast IP over the Broadcast Channel

MHP [1], clause 11.5.2 is included in the present document.

11.5.3 Support for IP over the Return Channel

MHP [1], clause 11.5.3 is included in the present document.

11.5.4 MPEG-2 Section Filter API

MHP [1], clause 11.5.4 is included in the present document.

11.5.5 Mid-Level Communications API

MHP [1], clause 11.5.5 is included in the present document, with the following notes and modifications.

The reference to annex R is to be read as referring to annex R of the present document.

11.5.6 Persistent Storage API

MHP [1], clause 11.5.6 is included in the present document.

11.6 Service information and selection APIs

11.6.1 Signalling-bound service information API

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

GEM terminal specifications shall provide any needed service information API specific to their signalling.

11.6.2 Service selection API

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

11.6.3 Tuning API

GEM terminal specifications for packaged media targets are not required to support a tuner. Except in the case of a packaged media GEM terminal specification that does not support a tuner, the definitions and requirements of MHP [1] clause 11.6.3 apply in the present document, with the following notes and modifications.

The reference to the `DvbLocator` class does not apply to the present document for GEM terminal specifications that do not include the MHP definition of the functional equivalent named "Content Referencing" as defined in clause 15.6, "Functional Equivalents".

The reference to MHP [1], clause 11.7.6 is to be read as referring to clause 11.7.6 of the present document.

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.

GEM terminal specifications may define any needed conditional access API.

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 for GEM terminal specifications that do not include the MHP definition of the functional equivalent named "SI" as defined in clause 15.6, "Functional Equivalents". However, GEM terminal specifications shall provide a mapping of the protocol independent SI API onto their SI signalling, as specified in annex O of the present document.

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: Support of the Xlet properties "`dvb.org.id`", "`dvb.app.id`" and "`dvb.caller.parameters`" is required.

In MHP [1], clause 11.7.1.1, the reference to the DVB-J application descriptor is to be interpreted as referring to clause 10.5.2, "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 S.

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

In GEM terminal specifications that include the MHP definition of the functional equivalent named "Carousel" in clause 15.6, "Functional Equivalents," full support of all properties defined for use with the method `AppAttributes.getProperty` in MHP [1] clause 11.7.2 is required. In GEM terminal specifications that do not include the MHP definition of this functional equivalent, support of these properties is optional.

Table 4 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 [4] 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 GEM terminal specifications that do not include these classes.

NOTE: The API from DAVIC defined in this clause is made optional for packaged media targets in clause 15.0, "General" because it relates to transport streams.

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 package `org.davic.net.dvb` is not required by the present document. GEM terminal specifications may define one or more subclasses of the DAVIC `Locator` class.

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 6.3.3.

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.

`NotAuthorizedInterface` and `NotAuthorizedException` are required to be present for GEM terminal specifications that include the section filter API. However, for GEM terminal specifications that do not include the MHP conditional access API, `NotAuthorizedException` shall not be generated by the platform.

NOTE: In such terminal specifications, there is consequently no mechanism for obtaining a platform object that implements `NotAuthorizedInterface`.

Support for the `ObjectUnavailableException` and `ResourceException` classes is not required for GEM terminal specifications that do not include the optional support for "11.7.4 Basic MPEG concepts" in the entry for "common infrastructure" in Table 7 of clause 15.0, "General" in the present document.

Support for the exception `TuningException` is not required for GEM terminal specifications that do not include the MPEG-2 Section Filter API.

11.8 Security

MHP [1], clause 11.8 is included in the present document, with the following notes and modifications. For this clause, subclauses are only included as explicitly indicated.

11.8.1 Basic Security

MHP [1], clause 11.8.1 is included in the present document.

11.8.2 APIs for return channel security

MHP [1], clause 11.8.2 is included in the present document.

MHP 11.8.2 has text that requires the minimum set of cipher suites to have names beginning with "`SSL_`" rather than "`TLS_`". This language might be interpreted as disallowing these cipher suites to also be reported with a name beginning with "`TLS_`". GEM terminal implementations are not subject to such a restriction; the method `getSupportedCipherSuites()` on the concerned classes may return cipher suite names beginning with "`TLS_`" in addition to the names starting with "`SSL_`" that are required to be returned.

MHP clause 11.8.2's text concerning the return value of the method `getSupportedCipherSuites` in four classes in the `javax.net.ssl` package has been interpreted in different ways. For GEM terminal specifications, this is to be read as requiring that these methods return at least the following constants:

```
SSL_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_WITH_DES_CBC_SHA
```

NOTE 1: `SSL_NULL_WITH_NULL_NULL` is not required to be returned. This cipher suite is used internally in the TLS/SSL protocol negotiation. Support of this protocol does not require that it be made available to applications via the getSupportedCipherSuites methods..

NOTE 2: If GEM terminal specifications add cipher suites in addition to the ones above, it may be appropriate for them to require the more current names beginning with `TLS_`, rather than names beginning with `SSL_`.

11.8.3 Additional permissions classes

MHP [1], clause 11.8.3 is included in the present document, with the following notes and modifications.

For GEM terminal specifications that do not support the MHP definition of the functional equivalent named "conditional access", the package `org.dvb.net.ca` is not required.

For GEM terminal specifications that do not support the MHP definition of the functional equivalent named "SI", the class `org.dvb.net.tuning.DvbNetworkInterfaceSIUtil` is not required.

For GEM terminal specifications that do not support the 11.6.3, "Tuning API" listed in Table 7 of the present document, the class `org.dvb.net.tuning.TunerPermission` is not required.

11.8.4 General Security Issues

MHP [1], clause 11.8.4 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 would return property values consistent with the MHP [1] interactive 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, GEM terminal specifications 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 terminal specifications that do not include the MHP definition of the functional equivalent named "Conditional Access" as defined in clause 15.6, "Functional Equivalents".

For GEM terminal specifications that do not support the "11.6.3 Tuning API" listed in Table 7 of the present document the class `org.dvb.net.tuning.TunerPermission` is not required by the present document and MHP [1] clause 11.10.2.6 does not apply.

GEM terminal specifications may define an alternative to MHP [1] clause 11.10.2.7, "`javax.tv.service.selection.SelectPermission`" in order to narrow the scope to apply only to broadcast services. Hence, applications will have the permissions needed to select all broadcast services in their own service context(s) unless denied by an entry in the permission request file.

NOTE: GEM terminal specifications may define types of service other than broadcast. The present document allows replacement of MHP clause 11.10.2.7 to only apply to broadcast services.

11.11 Content referencing

11.11.0 General

The following mapping shall be used between the types of locator defined in Table 6 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 6 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 and in the GEM terminal specification. 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.

For packaged media targets, support of locators to reference a transport stream are optional.

The term "DVB locators" is considered to refer to all valid locators as described in table 6. The informative note at the end of this clause referring to DVB-SI tables does not apply for GEM terminal specifications that do not include the MHP definition of the functional equivalent named "Content Referencing" as defined in clause 15.6, "Functional Equivalents."

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 6.

For the packaged media target, `NetworkCollection` and `Network` objects are never returned, as described in clause 14.8, "Locators and content referencing." For this reason, this clause does not apply to the packaged media target.

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 11.11.4.2 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()`.

The semantics for these methods described in MHP [1], clause 11.11.4.1 only apply if the GEM terminal specification requires a given method.

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().

The semantics for these methods described in MHP [1], clause 11.11.5 only apply if the GEM terminal specification requires a given method.

11.11.6 MPEG elementary stream

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

1) 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.

2) 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().

   The semantics for these methods described in MHP [1], clause 11.11.6 only apply if the GEM terminal specification requires a given method.

3) The methods on javax.media.MediaSelectControl shall accept locators for elementary streams as input in GEM terminal specifications where the functional equivalents for broadcast streaming video (defined in clause 7.2.2, "Video") and audio (defined in clause 7.2.1, "Audio") are carried as MPEG elementary streams.

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 6.

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 6.3.3) 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.

GEM terminal specifications may add rules similar to the above for locator types that they specify.

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.

The last paragraph of this clause is not included in the present document, and is replaced by the following:

- Applications that are eligible to be trusted shall be identified with an application_id from range of values allocated for signed applications in MHP [1] (see MHP [1] clause 12.5.1, table 12,"Value ranges for application_id"). Applications that are not eligible to be trusted shall be identified with an application_id from MHP's unsigned applications range. For an application with an application_id from the signed application range that requires signing to establish trust, if it is not signed it is considered to have failed authentication. An application with an application_id from the unsigned application range is treated as not eligible to be trusted even if the files might be transmitted with signatures.

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 "Extensions to MHP application signing framework" 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, shall:

- 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.

12.2 Authentication of applications

MHP [1], clause 12.2 is included in the present document with the following notes and modifications.

For profiles and/or targets that do not require compliance with the MHP definition of the functional equivalent named "application authentication" as defined in clause 15.6, "Functional Equivalents", GEM terminal specifications may define a functional equivalent for authentication of applications. This functional equivalent shall provide a mechanism to establish that code is sufficiently trusted to enable granting any of the permissions which a GEM application may request. Further, there shall be a mechanism to sign individual files or sets of files with a certificate that may be different from any that is used to determine the trust level of the application.

NOTE: Data files that are signed may have their certificates retrieved by applications using the `getSigners` methods described in clause P.2.1, "DSMCCObject".

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 1: In the case where the MHP application authentication mechanism is used, the exact file names, locations and syntaxes described in this clause must be supported. This includes the requirement in clause 12.4.3.1 that the last certificate in a CertificateFile be the root certificate.

NOTE 2: As specified in clause 12.2, "Authentication of applications" of the present document, an application authentication mechanism other than MHP's can be used to determine if an application is signed.

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

12.6.1 General principles

MHP [1], clause 12.6.1 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.

An alternative permission request mechanism may be used in order to request additional permissions over and above the ones defined in the present document. Any such alternative permission request mechanism shall not conflict with the MHP one and shall be completely defined in the GEM terminal specification with the following constraints:

- the permission requests shall be contained in a file encoded in XML format;
- the alternative permission request file shall be located in the directory that contains the initial file of the application;
- the formal public identifier of the alternative DTD shall identify the organization that specifies the alternative mechanism;
- the name of an alternative permission request file (PRF) shall be prefixed with a name identifying the GEM terminal specification, e.g. ocap.<application_name>.perm;
- the alternative DTD shall include at least all of the same XML elements and attributes as those defined in the DVB PRF.

Any additional permissions that are needed beyond those defined in GEM shall have their XML element names, global attribute names to be used with any element type, and local attribute names to be used with existing GEM defined element types prefixed by a string identifying the GEM terminal specification.

NOTE: Any local attributes on newly defined prefixed (qualified) XML element types that are used locally to those element types and are not intended to be used as global attributes should not be prefixed in accordance with the Per-Element-Type Partition defined by XML Namespaces [15], annex A.2, "XML Namespace Partitions."

12.6.2 Permission request file

MHP [1], clause 12.6.2 is included in the present document, with the following notes and modifications.

The following rules shall apply for the processing of PRFs by a GEM terminal:

1) If there is only a DVB PRF present, the GEM terminal shall use it.

2) If there is a platform-specific PRF (i.e. an OCAP PRF) present on the corresponding target platform (i.e. an OCAP terminal), then the GEM terminal shall use it exclusively.

3) If there is a platform-specific PRF present on a non-corresponding target platform, then it shall be ignored. In this case, if a DVB PRF is present, the GEM terminal shall use it.

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 [3] 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 still has to 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.

12.6.2.1 File encoding

MHP [1], clause 12.6.2.1 is included in the present document.

12.6.2.2 File integrity

MHP [1], clause 12.6.2.2 is included in the present document.

12.6.2.3 Example

MHP [1], clause 12.6.2.3 is included in the present document.

12.6.2.4 Permission request file name and location

MHP [1], clause 12.6.4 is included in the present document, with the following notes and modifications.

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.

12.6.2.5 Permission Request file

MHP [1], clause 12.6.2.5 is included in the present document.

12.6.2.6 Credentials

MHP [1], clause 12.6.2.6 is included in the present document, with the following notes and modifications.Since the "signature" and "certchainfileid" element documented in table 55 of MHP [1] is dependent on security files used in the MHP application authentication mechanism, stated elements may be modified depending on the application authentication mechanism used. Thus, GEM terminal specifications that do not use the MHP definition of the functional equivalent named "Application Authentication" as defined in clause 15.6, "Functional Equivalents". shall define an equivalent mechanism using the same XML syntax.

In table 55, in the "grantoridentifier" row, the "comments" column shall read: This element contains the 32 bit organization id identifying the grantor organization. The encoding of the CDATA attribute id of this element is "0x" followed by the encoding defined in clause 14.5, "Text encoding of application identifiers".

12.6.2.7 File Access

MHP [1], clause 12.6.2.7 is included in the present document.

12.6.2.8 CA API

MHP [1], clause 12.6.2.8 discusses the permissions for conditional access. As discussed in clause 11.10, "Java permissions" of the present document, the class `CAPermission` is not required to be present for GEM terminal specifications that do not include the MHP definition of the functional equivalent named "conditional access" as defined in clause 15.6, "Functional Equivalents"; for such GEM terminal specifications the policies specified in MHP [1] clauses 12.6.2.8.1 and 12.6.2.8.2 do not apply. However, the syntax of the `capermission` tag as specified in MHP [1], clause 12.6.2.8.3 shall be supported in all cases; if the MHP definition of "conditional access" is not included, it is to be silently ignored.

12.6.2.9 Application lifecycle control policy

MHP [1], clause 12.6.2.9 is included in the present document with the following notes and modifications.

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.

12.6.2.10 Return channel access policy

MHP [1] clause 12.6.2.10 is included in the present document with the following notes and modifications.

NOTE: 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.

12.6.2.11 Tuning access policy

MHP [1], clause 12.6.2.11 discusses the permissions for tuning access. As discussed in 11.10, "Java permissions" of the present document, the class `TunerPermission` is not required to be present for GEM terminal specifications that do not support the "11.6.3 Tuning API" listed in table 7 of the present document; for such GEM terminal specifications the policies specified in MHP [1] clauses 12.6.2.11.1 and 12.6.2.11.2 do not apply. If the MHP Function of "Tuning" is not included, it is to be silently ignored. However, the syntax of the `tuning` tag as specified in MHP [1], clause 12.6.2.11.3 shall be supported in all cases; if the tuning API is not included, it is to be silently ignored.

12.6.2.12 Service selection policy

MHP [1], clause 12.6.2.12 is included in the present document.

12.6.2.13 Media API access policy

MHP [1], clause 12.6.2.13 is included in the present document.

12.6.2.14 Inter-application communication policy

MHP [1], clause 12.6.2.14 is included in the present document.

12.6.2.15 User Setting and Preferences access policy

MHP [1], clause 12.6.2.15 is included in the present document.

12.6.2.16 Network permissions

MHP [1], clause 12.6.2.16 is included in the present document.

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

12.9.1 Certificate Revocation Lists

MHP [1], clause 12.9.1 is included in the present document, with the following notes and modifications.

For profiles and/or targets that do not require compliance with the MHP definition of the functional equivalent named "application authentication" as defined in clause 15.6, "Functional Equivalents." GEM terminal specifications are not required to implement certificate revocation lists; such GEM terminal specifications are therefore not required to comply with the requirements of this clause and its subclauses.

12.9.2 Root certificate management

For profiles and/or targets that do not require compliance with the MHP definition of the functional equivalent named "application authentication" as defined in clause 15.6, "Functional Equivalents," GEM terminal specifications are not required to implement root certificate management; such GEM terminal specifications are therefore not required to comply with the requirements of this clause and its subclauses.

NOTE: When application authentication mechanism defined in GEM terminal specification does not require a root certificate in the terminal, RCMM will not be necessary.

12.9.2.1 Introduction

MHP [1], clause 12.9.2.1 is included in the present document with the following notes and modifications.

MHP [1] specifies a root certificate management protocol that fulfils the requirements of GEM, but another protocol syntax and another distribution means are possible. Broadly speaking, GEM places requirements on both the semantics of a root certificate management message and requirements underlying its processing. GEM does not, however, mandate the syntax of the protocol message and its distribution means; this is left for GEM terminal specifications to define, either by using the MHP definition or by defining a functional equivalent.

12.9.2.2 Security of the root certificate replacement message

MHP [1], clause 12.9.2.2 is included in the present document with the following notes and modifications:

The reference to the `nextNbOfSignature` field is to be interpreted as referring to the version of `nextNbOfSignature` from the present document, as specified in clause 12.9.2.3.

12.9.2.3 Format of RCMM

RCMM encoding format as specified in MHP [1], clause 12.9.2.3 is not required for GEM terminal specifications. MHP [1], clause 12.9.2.3 shall apply to GEM terminal specifications that include the MHP definition of the functional equivalent named "RCMM" as defined in clause 15.6, "Functional Equivalents." GEM terminal specifications that do not include this MHP definition shall specify a functional equivalent that satisfies the requirements of this clause.

The unsigned RCMM message shall contain information sufficient to derive the following:

Table 5: Root Certificate Management Message description

Function	Type
thisUpdate	Date
nextNbOfSignatures	8-bit Unsigned Integer
addedCertificates	set of certificates
removedCertificates	set of certificate references

GEM-based terminal specifications are allowed to define additional fields in the RCMM message.

The signatures shall subsequently be computed on the whole content of the unsigned RCMM, including possible extensions defined by GEM-based terminal specifications.

thisUpdate: Date of issue of the message.

nextNbOfSignatures: This field is used to change the minimum number of valid signatures required for an RCMM message. This value will be applied to the next RCMMs not to itself.

addedCertificates: List of root certificates to be added in persistent storage.

removedCertificates: Reference of the root certificates to be removed from persistent storage.

12.9.2.4 Distribution of root certificate replacement messages

MHP [1], clause 12.9.2.4 is not required for GEM terminal specifications. MHP [1], clause 12.9.2.4 shall apply to GEM terminal specifications that include the MHP definition of the functional equivalent named "RCMM" as defined in clause 15.6, "Functional Equivalents." GEM terminal specifications that do not include this MHP definition shall specify a functional equivalent that satisfies the requirements of this clause.

RCMM messages shall be distributed to the MHP terminals in the broadcast MPEG Transport Stream. The distribution mechanism to be specified in a GEM terminal specification that do not support the MHP definition of the functional equivalent named "RCMM" as defined in clause 15.6, "Functional Equivalents" shall take into account the need to broadcast a set of historic RCMM messages in addition to the latest one. See MHP [1], clause 12.9.2.4.

12.9.2.5 RCMM Processing

MHP [1], clause 12.9.2.5 is included in the present document with the following notes and modifications:

References to RCMM messages named "dvb.rcmm.<x>" shall be interpreted as the set of historic RCMM messages as mentioned in clause 12.9.2.4 of the present document. References to RCMM message named "dvb.rcmm" shall be interpreted as the most recent RCMM message.

The reference to the `thisUpdate` field is to be interpreted as referring to the version of `thisUpdate` from the present document, as specified in clause 12.9.2.3.

The reference to the `nextNbOfSignature` field is to be interpreted as referring to the version of `nextNbOfSignature` from the present document, as specified in clause 12.9.2.3.

12.9.2.6 Example: Renewal of a root certificate

MHP [1], clause 12.9.2.6 is included in the present document.

12.9.3 Test certificates

MHP [1] clause 12.9.3 is included in the present document.

NOTE: The test certificates guarantee that there is a way to cause a GEM terminal to grant all permissions requested by an application. The handling of these test certificates is decided by the authors of a GEM terminal specification.

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, with the following notes and modifications.

For GEM terminal specifications that do not adopt the MHP definition of the functional equivalent named "application authentication" as defined in clause 15.6, "Functional Equivalents", GEM terminal specifications may specify different Platform minima than what is described in clause 12.12 of MHP [1].

NOTE 1: When application authentication mechanism defined in GEM terminal specification does not require a root certificate in the terminal, the platform minima concerning the root certificate described in clause 12.12 of MHP [1] do not apply.

NOTE 2: CRL and RCMM related minima do not apply for Packaged Media profile.

13 Graphics reference model

13.0 General

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

13.1 Supported graphics resolutions

MHP [1], clause 13 contains references to the platform minima in annex 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 Aspect Ratio Support

MHP [1] clause 13.3.7 mandates the 14:9 aspect ratio. This clause is not required for GEM terminal specifications.

13.3 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. However, as described in clause 7.2.2 a GEM terminal specification shall include some mechanism for delivering MPEG-2 audio and video programming in standard definition. For these formats, background JMF players shall be created. Thus, the last paragraph of clause 13.4.1 applies to this player.

13.4 Subtitles

As signalling to support subtitles is not required by the present document, MHP [1], clause 13.5 only applies to GEM terminal specifications for which subtitling is available. If subtitling signalling is available, the presentation of subtitles shall follow the model specified in MHP [1], clause 13.5. Further, the introduction to subtitles contained in MHP [1] clause 13.1 and the components related to subtitles in figure 17 in MHP [1] clause 13.2.1 only apply to GEM terminal specifications that include support for subtitling.

NOTE 1: 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 optional.

NOTE 2: See also clause N.1, "Active Format Definition".

14 System integration aspects

14.1 Namespace mapping

The present document does not mandate any particular format for locators. Note, however, that clause 6.3.3 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: GEM terminal specifications may extend the allowed XML notation by permitting other character encodings, if this extension is explicitly stated in a GEM terminal specification. If no encoding is specified in an XML file, however, the default shall be UTF-8.

14.4 Network signalling (error behaviour)

MHP [1], clause 14.4 is included in the present document.

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 Filename requirements

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 6.3.3.

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

14.8 Locators and content referencing

Table 6 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 GEM terminal specifications shall 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 6: Addressable entities, locators and their text representation

Entity	Text Representation
Required for all targets:	
Service	locator text representation shall be defined
Service domain	locator text representation shall be defined
MPEG Elementary Stream	locator text representation shall be defined
File	"file:", "http:" and "https:" URLs, as referred to in MHP [1], clause 14.8 locator text representation shall 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 shall be defined for directories located within a Service domain
Required for broadcast targets only:	
Transport stream	locator text representation shall be defined
Network	no standardized text representation required
Program Event	locator text representation shall be defined
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 independent locators. A standardized textual representation of transport independent locators is not required by the present document.

NOTE: In MHP, transport dependent locators are represented as instances of `org.davic.net.dvb.DvbNetworkBoundLocator`. Transport independent locators are represented by instances of `org.davic.net.dvb.DvbLocator` which are not transport dependent as defined above. The textual representation of transport independent locators is defined by MHP [1] clause 14.1, "Namespace mapping (DVB Locator)". The textual representation of transport dependent locators is implementation-specific.

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 include the MHP requirements in clause 14.10 as described in clause 15.6, "Functional Equivalents".

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 7 is organized by the two classes of GEM terminal specification targets, broadcast and packaged media. For each target, there are two profiles, the enhanced profile (abbreviated "E.P." in table 7) and the interactive profile (abbreviated "I.P." in the table). These two profiles mirror the two profiles in MHP [1].

Table 7: Platform profile definitions

Area	Specification	Broadcast Targets		Packaged Media Targets	
		E.P.	I.P.	E.P.	I.P.
Static Formats					
Bitmap pictures	7.1.1.3, "PNG" + 15.1, "PNG - restrictions"	M	M	M	M
	7.1.1.3, "PNG" without restrictions	-	-	-	-
	7.1.1.4, "GIF"	-	-	-	-
	7.1.2, "MPEG-2 I-Frames"	M	M	M	M
	7.1.1.2, "JPEG" + 15.3, "JPEG - restrictions"	M	-	M	-
	7.1.1.2, "JPEG" without restrictions	-	M	-	M
Audio clips	7.1.4, "Monomedia format for audio clips"	M	M	M	M
Video drips	7.1.3, "MPEG-2 Video "drips""	M	M	M	M
Text encoding	7.1.5, "Monomedia format for text"	M	M	M	M
Media Streaming formats					
Video	7.2.2, "Video"	M	M	M	M
Audio	7.2.1, "Audio"	M	M	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	O	O
Downloadable	7.4, " Downloadable fonts"	M	M	M	M
Broadcast channel protocols					
	6.2.2, "MPEG-2 sections"	M	M	M	M
	6.2.5, "Object carousel"	M	M	-	-
	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	6.3.3, "Transmission Control Protocol" 6.3.2, "Internet Protocol"	-	M	-	M
UDP/IP	6.3.2, "Internet Protocol" 6.3.9, "User Datagram Protocol"	-	M	-	M
HTTP	6.3.7.1, "HTTP 1.1"	-	O	-	O

Area	Specification	Broadcast Targets		Packaged Media Targets	
		E.P.	I.P.	E.P.	I.P.
DSMCC-UU RPC	6.3.4, "UNO-RPC" 6.3.5, "UNO-CDR" 6.3.6, "DSM-CC User to User"	-	O	-	-
DNS	6.3.10, "DNS"	-	M	-	M
DVB-J					
Core	11.3, "Fundamental DVB-J APIs"	M	M	M	M
Presentation	11.4, "Presentation APIs" (note applies in all cases)	M	M	M	M
Data Access	11.5.1, "Broadcast Transport Protocol Access API"	M	M	M	M
	11.5.2, "Support for Multicast IP over the Broadcast Channel"	O	Ro	-	-
	11.5.3, "Support for IP over the Return Channel"	-	M	-	M
	11.5.4, "MPEG-2 Section Filter API"	M	M	O	O
	11.5.5, "Mid-Level Communications API" as modified by 11.5, "Data access APIs"	O	M	O	M
	11.5.6, "Persistent Storage API"	M	M	M	M
Service Information and Selection	11.6.1, "Signalling-bound service information API"	M	M	M	M
	11.6.2, "Service selection API"	M	M	M	M
	11.6.3, "Tuning API"	M	M	-	-
	11.6.4, "Conditional access API"	-	-	-	-
	11.6.5, "Protocol independent SI API"	M	M	M	M
Common Infrastructure	11.7.1, "APIs to support DVB-J application lifecycle"	M	M	M	M
	11.7.2, "Application discovery and launching APIs"	M	M	M	M
	11.7.3, "Inter-application communication API"	M	M	M	M
	11.7.4, "Basic MPEG concepts"	M	M	-	-
	11.7.5, "Resource notification"	M	M	M	M
	11.7.6, "Content referencing"	M	M	M	M
	11.7.7, "Common error reporting"	M	M	M	M
Security	11.8.1, "Basic Security"	M	M	M	M
	11.8.2, "APIs for return channel security"	-	M	-	M
	11.8.3, "Additional permissions classes"	M	M	M	M
Others	11.9.1, "Timer support"	M	M	M	M
	11.9.2, "User settings and preferences API"	M	M	M	M
	11.9.3, "Profile and version properties"	M	M	M	M

Key	
-	Not applicable/Not required
O	Optional feature in the GEM terminal
Ro	Recommended optional feature in the GEM terminal
M	Mandatory feature in the GEM terminal

If a GEM terminal specification wishes to include APIs, signalling or behaviours defined in MHP [1] as functional equivalents, it shall do so by referencing the corresponding entry in clause 15.6, "Functional Equivalents" and not by directly referencing MHP.

15.1 PNG - restrictions

MHP [1], clause 15.1 is included in the present document.

15.2 Minimum media formats supported by DVB-J APIs

MHP [1], clause 15.2 is included in the present document, with the following notes and modifications.

Support for subtitles is optional.

Media formats in the table entitled "Media type support required in Enhanced and Interactive Broadcast profile 1" may be replaced with functional equivalents as permitted by clause 15.6, "Functional Equivalents."

15.3 JPEG - restrictions

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

15.4 Locale support

MHP [1], clause 15.4 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 locale 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.

NOTE: This is the a convention adopted by the Java2D API. See the specification of `java.awt.Font.getSize()`. As a result of this convention, the return value of `getSize()` gives the size in pixels for a screen device.

15.6 Functional Equivalents

The table 8 lists the set of mandatory and optional functional equivalents. Those elements marked mandatory shall be defined in a GEM terminal specification. Those marked optional may be defined in a GEM terminal specification.

If a GEM terminal specification wishes to include technology defined in MHP [1] that is not required by GEM, it may do so only for one or more named MHP definitions of functional equivalents listed in column 1 in table 8 and shall be conformant with the clauses for that element referenced by referenced by table 8.

Each reference to a clause in MHP [1] in this table refers to the original MHP technology, including all references within MHP [1]. For references in the "MHP definition" column, the rule about internal references in clause 4.2.1 of the present document does not apply.

NOTE 1: This means that references in the "MHP definition" column refer to the original MHP clauses unmodified by GEM.

Each functional equivalent is identified by a name in the leftmost column of table 8.

For each mandatory named functional equivalent, the GEM terminal specification shall explicitly indicate that it adopts the entire MHP Definition as referenced in table 8 or shall specify a replacement as required by the corresponding GEM clause(s) in table 8.

For each optional named functional equivalent, the GEM terminal specification shall explicitly indicate whether it is supported. If supported, the GEM terminal specification shall indicate that it adopts the entire MHP Definition as referenced in table 8 or shall specify a replacement as required by the corresponding GEM clause(s) in table 8.

Table 8 is organized by the two classes of GEM terminal specification targets, broadcast and packaged media. For each target, there are two profiles, the enhanced profile (abbreviated "E.P." in the table) and the interactive profile (abbreviated "I.P." in the table). These two profiles mirror the two profiles in MHP [1].

Table 8: Functional Equivalents

Name	GEM Clause(s)	MHP Definition	Broadcast Targets		Packaged Media Targets	
			E.P.	I.P.	E.P.	I.P.
Arch	5, "Basic architecture"	MHP [1], clause 5, "Basic Architecture"	O	O	O	O
Carousel	6.2.5, "Object carousel" See also 11.7.2, "Application discovery and launching APIs"	MHP [1], clause 6.2.5, "DSMCC User-to-user Object Carousel" as modified by clause 15.6.1.1, "Carousel"	M	M	M	M
IP MPE	6.2.6, "Protocol for delivery of IP multicast over the broadcast channel"	MHP [1], clause 6.2.6, "DVB Multiprotocol Encapsulation"	O	O	O	O
SI	6.2.9, "Service information"	MHP [1], clause 6.2.9, "DVB Service Information"	M	M	M	M
	11.6.1, "Signalling-bound service information API"	MHP [1], clause 11.6.1, "DVB Service Information API" MHP [1], clause 11.11.3, "Bouquet"				
	Annex O, "Integration of the JavaTV SI API"	MHP [1], annex O, "Integration of the JavaTV SI API"				
Broadcast IP signalling	6.2.10, "IP signalling"	MHP [1], clause 6.2.10, "IP signalling"	O	O	O	O
Audio	7.2.1, "Audio"	MHP [1], clause 7.2.1, "Audio"	M	M	M	M
Video	7.2.2, "Video"	MHP [1], clause 7.2.2, "Video"	M	M	M	M
Subtitles	7.2.3, "Subtitles"	MHP [1], clause 7.2.3, "Subtitles"	O	O	O	O
	11.4, "Presentation APIs" - classes related to subtitles	MHP [1], clause 11.4.2.5.1, classes related to subtitles				
Audio Clips	7.1.4, "Monomedia format for audio clips"	MHP [1], clause 7.1.4	M*	M*	M	M
Resident Fonts	7.3, "Resident fonts"	MHP [1], clause 7.3	M*	M*	M	M
Downloadable Fonts	7.4, "Downloadable fonts"	MHP [1], clause 7.4	M*	M*	M	M
Application Signalling	10.2, "Program specific information" 10.4, "Application description" 10.5, "DVB-J specific application description"	MHP [1], clause 10.2, "Program Specific Information" as modified by clause 15.6.1.2, "Application Signalling" MHP [1], clause 10.3, "Notation" MHP [1], clause 10.4, "Application Information Table" MHP [1], clause 10.5, "Application Identification" MHP [1], clause 10.6, "Control of application life cycle" MHP [1], clause 10.7, "Generic descriptors" as modified by clause 15.6.1.3, "Application Name Descriptor" MHP [1], clause 10.8, "Transport protocol descriptors" MHP [1], clause 10.9, "DVB-J specific descriptors" MHP [1], clause 10.11, "Constant values" as modified by clause 10.6, "Constant Values"	M	M	M	M
	11.7.2, "Application discovery and launching APIs"	MHP [1], clause 11.7.2, "Application discovery and launching APIs"				

Name	GEM Clause(s)	MHP Definition	Broadcast Targets		Packaged Media Targets	
			E.P.	I.P.	E.P.	I.P.
Application Authentication	12.2, "Authentication of applications" NOTE 1: See also text in clause12.6.2.6, "Credentials" and text in clause 12.9, "Certificate management" NOTE 2: See also text in clause 12.9.2, "Root certificate management."	MHP [1], clause 12.2, "Authentication of applications"	M	M	M	M
	12.9.1, "Certificate Revocation Lists"	MHP [1] clause 12.9.1, "Certificate Revocation Lists"				
Conditional Access	11.4, "Presentation APIs," classes related to conditional access 11.6.4, "Conditional access API"	MHP [1], clause 11.4.2.5.1, classes related to conditional access MHP [1], clause 11.6.4, "Conditional Access API" MHP [1], clause 14.10, "CA system"	O	O	O	O
Content Referencing	11.7.6, "Content referencing"	MHP [1], clause 11.7.6, "Content Referencing"	M	M	M	M
	11.11.11, "Methods working on many locator types"	MHP [1], clause 11.11.11, "Methods working on many Locator types"				
	14.1, "Namespace mapping"	MHP [1], clause 14.1, "Namespace mapping (DVB Locator)"				
	14.9, "Service identification"	MHP [1], clause 14.9, "Service identification"				
Graphics Resolution	D.2, "Horizontal resolution"	MHP [1], clause D.3.4.2, "Horizontal resolution"	M	M	M	M
	G.1.1, "Device resolution for Standard Definition"	MHP [1], clause G.1.1, "Device capabilities"				
	G.4, "Resident fonts and text rendering"	MHP [1], clause G.4, "Resident fonts and text rendering"				
Text Wrapping	D.3, "Text wrapping setting is true"	MHP [1], clause D.3.7.2, "Text wrapping setting is true"	M	M	M	M
Minimum CLUT	G.1.3, "Minimum Colour Lookup Table"	MHP [1], clause G.1.5, "Colour capabilities"	M	M	M	M
RCMM	12.9.2, "Root certificate management"	MHP [1], clause 12.9.2, "Root Certificate Management"	M	M	M	M
Active Format Descriptor	N.1, "Active Format Definition"	Described in the present document, clause N.1.1, "MHP Signalling for Active Format Definition"	M	M	M	M

Key	
M	Functional equivalent required
M*	MHP definition of the functional equivalent required
O	Optional feature, no functional equivalent required
-	Not required/Not applicable

NOTE 2: Features are labelled "O" if they could reasonably be applied to the given target and/or profile. This implies that if this feature is implemented, it will be done according to the specification provided by the present document.

15.6.1 Modifications to MHP Definitions of Functional Equivalents

In certain cases, it may be necessary to slightly modify the MHP definition of a functional equivalent. A GEM terminal specification that adopts the MHP definition of a functional equivalent may modify it only as described in this clause.

15.6.1.1 Carousel

15.6.1.1.1 NSAP Address

The DVB Carousel NSAP Address structure is defined in table B.26 in clause B.2.3.7.2 MHP [1]. It may be modified by replacing the definition of `specifierType`, `specifierData`, and `dvb_service_location` structures with structures specified in the GEM terminal specification.

NOTE: This may be necessary because the `dvb_service_location` is based on `transport_stream_id`, `original_network_id` and `service_id` values that come from DVB SI.

15.6.1.1.2 Content type descriptor

MHP [1] clause B.2.3.4 defines rules for determining the MIME type of a file based on the file extension, if a content type descriptor is not present in a file object. GEM terminal specifications may define another mechanism to determine the MIME type of a file that takes precedence over the mapping based on the extension portion of the file name, provided that a correctly built MHP object carousel will be processed in a way compatible with MHP.

15.6.1.1.3 Application Icons Descriptor

GEM terminal specifications using the MHP functional equivalent of application signalling may define new application types. If they do, they may define icon locator semantics for the application types they introduce in the application icons descriptor, which is defined in MHP [1], clause 10.7.4.2.

15.6.1.2 Application Signalling

15.6.1.2.1 Transport protocol descriptor

GEM terminal specifications using the MHP definition of the functional equivalent named "application signalling" are not required to include the protocol_ids and corresponding selector byte definitions found in MHP [1] clause 10.8.1, "Transport protocol descriptor". If they do not include these MHP definitions, they shall define additional `protocol_id`s and corresponding selector bytes to replace these. New `protocol_id` values shall be registered with the DVB.

15.6.1.3 Application Name Descriptor

GEM terminal specifications using the MHP definition of the functional equivalent named "application signalling" may modify the definition of the `application_name_char` field of the application name descriptor defined in MHP [1] clause 10.7.1, as follows: The encoding of this string may be changed, provided that all strings that start with the value 0x20 are interpreted as per the MHP definition of this functional equivalent.

NOTE 1: The MHP definition, taken from annex A of ETS 300 468 [18], requires that string that start with the value 0x20 be interpreted as an ASCII string.

NOTE 2: See also clause A.1.2, "Application name descriptor" in the present document. For simplicity, GEM terminal specifications may wish to limit the encoding of this field to the required ASCII encoding an UTF8.

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.

MHP [1], clause 16.2.2 requires the constant `java.awt.event.KeyEvent.CHAR_UNDEFINED`. In GEM terminal specifications, this constant is required to be present but its value may be either 0 or 0xffff.

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

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

NOTE: With the exception of the issue discussed in clause A.3, the agreed errata to Java TV present in the present document and in MHP have been incorporated into Java TV 1.1 [12]. Implementers of GEM terminal specifications based on Java TV 1.1 need not consider errata to Java TV listed in GEM or MHP.

A.1 Errata to MHP

The following errata to MHP [1] shall apply, with the exception of clause 3.2.2, "Issue 4195".

A.1.1 MHP Errata Document

The errata in tm2971r2 [7] shall apply, with the following exceptions:

- The change in tm2971r2 clause 4.15.1 (issue 4293, "VideoFormatControl" places an extra requirement on MHP terminals that is optional for GEM terminal specifications.
- The change in tm2971r2 clause 4.18.6 (issue 4258, "AppProxy destroyed behavior") is optional for GEM terminal specifications.
- The change in tm2971r2 clause 4.20.1 (issue 4289, "DVBTextLayoutManager") is optional for GEM terminal specifications.

The change in tm2971r2 clause 4.7.6 (issue 4049, "HScene Issue") is optional for GEM terminal specifications.

The change in tm2971r2 clause 4.7.6 (issue 4258, "App Proxy") should be replaced with the following for GEM terminal specifications.

In the class description for `AppProxy.DESTROYED`, the following text:

> The application is in the destroyed state. This state is transient and entry to this state shall be followed with a transition to the `NOT_LOADED` state almost immediately. It shall be possible to re-start the application after the transition to the `NOT_LOADED` state.

is replaced with;

An `AppProxy` for a terminated application is no longer usable, An `AppProxy` instance should be considered invalid, once it was in the `AppProxy.DESTROYED` state.

NOTE: If a terminated application needs to be restarted, an `AppProxy` for a new application instance may be obtained from the AppsDatabase.

In the class description for `org.dvb.media.VideoFormatControl` the following text:

> This provides a means for applications to get information associated with the format and aspect ratio of the video being presented to the user. This control will only be available for Players presenting MPEG-2 video streams.

is replaced with:

> This provides a means for applications to get information associated with the format and aspect ratio of the video being presented to the user. This control will only be available for Player instances presenting video formats.

A.1.2 Application name descriptor

In clause 10.7.4.1 "Application name descriptor", the description of the `application_name_char` field shall be considered to read as follows:

This field carries a string (not null terminated) of characters encoded in accordance with annex A of EN 300 468 [10] with the modification that the first byte of a text field having the value 0x16 shall be interpreted as meaning UTF-8 encoding as defined by ISO/IEC 10646 [11].

NOTE: The value of 0x16 was agreed in the July 2004 meeting of the DVB-GBS group but has not yet been included in a published version of EN 300 468 [10] or the MHP specification.

A.1.3 Java TV Overview - Enumerated Types

MHP [1] shall be considered to include a clarification to Java TV as follows.

NOTE: An MHP specification issue has been agreed by DVB-TAM, and will result in this language being present in a future version of MHP.

The Java TV overview is considered to be supplemented with the following clause after the paragraph entitled "Callback Threads":

Enumerated Types

Java TV defines the following enumerated types:

- javax.tv.service.navigation.DeliverySystemType;
- javax.tv.service.guide.ProgramScheduleChangeType;
- javax.tv.service.ServiceInformationType;
- javax.tv.service.ServiceType;
- javax.tv.service.SIChangeType;
- javax.tv.service.SIRequestFailureType;
- javax.tv.service.navigation.StreamType.

Platforms including Java TV may define additional values for these types through subclassing. Applications written to Java TV or to any specification that includes Java TV should not rely on methods using values of these types being constrained to the set of values defined as constants in Java TV; values unknown to the application author may be returned.

Specification authors are advised not to define new values that semantically overlap with the values defined by Java TV or elsewhere.

A.1.4 org.dvb.io.persistent.FileAttributes class

In org.dvb.io.persistent.FileAttributes, the following constructor shall be considered to be present;

```
/**
  * Constructor.
  *
  * @param expiration_date an expiration date or null
  * @param p the access permissions to use
  * @param priority the priority to use in persistent storage
  *
  */
  public FileAttributes(Date expiration_date, FileAccessPermissions p,
                int priority) {}
```

A.1.5 org.davic.media.LanguageControl class

In clause A.6.6.4, "LanguageControl":

a) insert a new heading immediately following this heading as follows:

A.6.6.4.1 Class description

b) Insert the following additional clause:

A.6.6.4.2 selectLanguage(String)

The following additional text shall be considered to form part of the description of this method;

When this method returns normally, the language will have been synchronously selected.

A.1.6 `application_id` range for unsigned applications

In clause 10.5.1, "Encoding", table 12, "Table 12: Value ranges for `application_id`":

a) in the row "Application_ids for unsigned applications", in the cell "application_id values", the text "`0x0000...0x3fff`" shall be considered to read "`0x0001...0x3fff`".

b) a new row shall be considered to be added immediately after the heading as follows:

`0x0000` shall not be used.

A.1.7 DripFeedPermission name parameter

In annex N, "Streamed Media API Extensions," for the class `DripFeedPermission`, the documentation of both constructors shall be considered to include the following text:

> The value of the name parameter may be ignored by implementations, and the value passed to the superclass may be a non-empty string even when this constructor is invoked with an empty string for the name parameter. Therefore, the `getName` method may return the same non-empty string regardless of the value passed in the constructor.

A1.8 Data rate for return channel interface

In the class `org.dvb.net.rc.RCInterface` the description of the return value of the method `getDataRate` should read:

Returns:

a data rate in Kbit/s or -1 where this is not available.

A.2 Errata to DAVIC

The following errata to DAVIC [13] shall apply:

A.2.1 org.davic.media.MediaTimeEventControl - deregistering listeners

The following shall be considered to be added to the description of the main interface description of `org.davic.media.MediaTimeEventControl`:

> It is an implementation's responsibility to deregister any registered instances of MediaTimeEventListener at an appropriate time, e.g. when the Xlet is destroyed.

The following shall be considered to be added to the description of the `notifyWhen(org.davic.media.MediaTimeEventListener i, long mediaTime, int id)` method:

> When this method is called with a listener, an id and a negative mediaTime as arguments, the listener is deregistered for the negated value of the corresponding mediaTime parameter and matching id.
>
> The availability of this deregistration feature on the platform may be indicated via the existence of the system property.
>
> Calling this method with a value that does not match a previously registered positive media time shall have no effect.

NOTE 1: When this method is called with a `mediaTime` value of 0 the result is implementation dependent.

NOTE 2: When an application calls `notifyWhen` more than once with the same `mediaTime` and `id`, it is implementation dependent if more than one event is generated and whether multiple deregistrations will be required.

The following shall be considered to be added to the description of the `notifyWhen(org.davic.media.MediaTimeEventListener i, long mediaTime)` method:

> When this method is called with a listener and a negative mediaTime as arguments, the listener is deregistered for the negated value of the corresponding mediaTime parameter.
>
> The availability of this deregistration feature on the platform may be indicated via the existence of a system property.

NOTE 3: A deregistration via this method is equivalent to calling `notifyWhen(org.davic.media.MediaTimeEventListener i, long mediaTime, int id)` with an `id` value of 0.

NOTE 4: Although this class is not required by MHP or by any profile of GEM, GEM terminal specifications may include it as a mandatory or optional element. For example, BD-J is known to require this class.

A.2.2. org.davic.resources.ResourceClient.requestRelease

The following text in the description of the `requestRelease()` method:

> A call to this operation informs the `ResourceClient` that another application has requested the resource accessed via the proxy parameter.

shall be considered to be replaced with:

> A call to this operation informs the `ResourceClient` that another `ResourceClient` instance has requested the resource accessed via the proxy parameter.

A.3 Additional MHP requirements on Java TV

A.3.1 javax.tv.util.TVTimerSpec

NOTE 1: MHP [1] clause A.5.2.2, "deschedule(TVTimerSpec)" requires that the argument be an instance returned from the method TVTimer.schedule. This is not a requirement of Java TV, but GEM terminal implementations are required to obey this MHP-defined behavior.

NOTE 2: Java TV 1.1 [12] requires that the argument to deschedule(TVTimerSpec) be the instance passed as argument to the method TVTimer.schedule. Implementations of GEM terminal specifications based on Java TV 1.1 are required to obey this JavaTV-defined behavior. This may be implemented in a way consistent with the MHP requirements by implementing TVTimer.schedule such that it always returns the instance passed in as argument. Note that Java TV's specification of TVTimer.schedule allows the implementation to modify the instance passed in as argument, e.g. to account for timer granularity.

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 GEM terminal specifications 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 shall 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).

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 shall include a mechanism for delivering a hierarchical file system within a service domain. It shall be possible to construct a locator that refers to files and directories in this hierarchy. The file system delivery mechanism shall 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	500
Maximum Directory Nesting	20 levels
Maximum File Length	2^{27}-1 bytes
Caching	See clause B.2.1.1

Filesystems may be either case-sensitive or "case preserving." GEM applications shall be written to work with both of these. "case preserving" is defined in MHP [1], clause 14.6.1. GEM terminal specifications may require case-sensitive filesystems.

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.

NOTE: For the packaged media target, the contents of the filesystem usually does not change. For this reason, explicit signalling of the transparency level is not required."

B.2.2 Filesystem updates

For broadcast targets, 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 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	string
event_id	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. For broadcast targets, the signalling shall contain information sufficient to derive the following. For the packaged media profile, it shall be possible to specify trigger events with a payload as described; the distinction between "do it now" events and time-based events is not meaningful and need not be encoded.

NOTE 1: In MHP, applications cannot detect the signalling underlying a trigger event.

Table B.5: Trigger event

Function	Type
event_name	string
is_do_it_now	flag
timebase_value	32 bit unsigned integer
payload	byte array

event_name: The name of the trigger event. The signalling shall support trigger names up to 32 characters long containing the ASCII characters "a".."z", "A".."Z", "0".."9" and "-".

NOTE 2: This may be signalled as an event_id that maps to a trigger identification within a trigger object, as defined in clause B.4.1 Trigger object.

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.

timebase_value: A value that identifies a point in time in a timebase signalled with a service. This is implemented as an MPEG NPT value in the DSMCC signalling.. For "do it now" events, this value is ignored.

NOTE 3: 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 4: A trigger event corresponds to a DSMCC section carrying a stream event descriptor.

B.4.2.1 Extrapolation of timebase values

GEM terminal specifications for broadcast targets shall be written such that, for broadcasts confirming to appropriate broadcast norms and specifications and absent reception errors, any extrapolation of timebase 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".

GEM terminal specifications for packaged media targets shall require detecting triggers associated with the currently played media stream.

B.4.2.2 Monitoring of trigger events

For each GEM application, GEM terminal specifications for broadcast targets shall require monitoring 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 Other informative references

[19] ETSI TS 102 812: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.1".

[20] W3C Recommendation 20060816: "XML Namespaces".

NOTE: Available at http://www.w3.org/TR/REC-xml-names.

[21] Application Definition Blu-ray Disc Format - BD-J Baseline Application and Logical Model Definition for BD-ROM, March 2005.

NOTE: Available at http://www.blu-raydisc.com/ under "Technical Info", "Public Specifications".

[22] ETSI TR 101 162: "Digital Video Broadcasting (DVB); Allocation of Service Information (SI) codes for DVB systems".

[23] ETSI ETS 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

Annex D (normative): Text presentation

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

D.1 Font Technology

MHP [1], clause D.2.2.1 references clause 7.4. For the present document this reference is to be interpreted as referring to clause 7.4, "Downloadable fonts" in the present document.

D.2 Horizontal resolution

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 shall be specified in GEM terminal specifications.

D.3 Text wrapping setting is true

MHP [1], clause D.3.7.2 may not apply for non-western languages. GEM terminal specifications that support languages for which MHP [1], clause D.3.7.2 does not apply shall provide a functional equivalent.

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 notes and modifications detailed in the following clauses.

G.1.1 Device resolution for Standard Definition

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

G.1.2 Device Resolution for Standard Definition

The first bullet point of MHP [1], clause G.1.2 mandates support for a number of decoder format conversions. However, GEM terminal specifications for packaged media targets only require support for DFC_PROCESSING_FULL.

G.1.3 Minimum Colour Lookup Table

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 shall be provided. It shall support at least 139 opaque colours, 48 colours at 30 % transparency, and 1 colour at 100 % transparency. Where the receiver cannot implement the "ideal" value of 30 % transparency it shall replace it with the nearest value of semi-transparency it can implement. Semi-transparency shall not be approximated as either 0 % or 100 % transparency. It is not required for a GEM terminal specification to define these colours; for example, a GEM terminal specification requiring at least 16 bits per pixel meets the requirement of supporting at least 139 opaque colours.

A GEM terminal specification based on a CLUT system should define 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.

The second column of table G.2 in clause G.4.1 is informative in MHP [1] as amended by tm2971r1 [7], and it may not apply to all GEM terminal specifications. For clarity GEM terminal specifications may specify an equivalent of these values, though if they do not this does not imply that the second column of table G.2 is accurate for these specifications.

MHP [1], clause G.4.1 discusses the built-in font. As described in clause 7.3, "Resident fonts" in the present document, the resident font might not be built into the terminal in all cases.

G.5 Input events

MHP [1], clause G.5 is included in the present document with the following notes and modifications:

GEM-based terminal specifications are allowed to make optional the VK_TELETEXT and VK_COLORED_KEY_3 input events.

NOTE 1: Remote control devices should be designed such that the mechanism for generating each required event is obvious to typical end-users. Hence, application authors may assume that the required keys can be generated without recourse to non-obvious input gestures.

NOTE 2: See also the solution to issue 4106 in tm2971r1 [7]. This requires a meaningful representation of how a given supported keycode is generated

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 [3]. In this case, such downloaded software shall implement a policy that is consistent with the requirements of the present document, e.g. MHP [1], clause G.5.

GEM terminal specifications should not define additional input event codes where MHP [1] defines an appropriate VK_ code.

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 section filtering. As an MHP AIT is not required, this entry does not apply to GEM terminal specifications that do not include the MHP definition of the functional equivalent named "Application Signalling" as defined in clause 15.6, "Functional Equivalents." 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. This does not apply to GEM terminal specifications that do not support the MHP definition of the functional equivalent named "Conditional Access" as specified by MHP [1].

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, with the following notes and modifications.

Clause 16.2, "DVB-J constants" alters the definition of the constant java.awt.event.KeyEvent.CHAR_UNDEFINED. The return value of org.dvb.event.UserEvent.getKeyChar() is defined in terms of CHAR_UNDEFINED. The value of getKeyChar() for this method is not defined in an interoperable way. In no case should an interoperable application rely on the return value of getKeyChar(); further, the value returned for an undefined key is not defined interoperably and may vary across platform implementations.

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 (normative): SI Access API

MHP [1] annex M is not included in the present document.

NOTE: The MHP definition of the functional equivalent named "SI" in clause 15.6, "Functional Equivalents." references MHP [1], annex M.

Annex N (normative): Streamed media API extensions

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

References to "720 x 576" frames are to be read as referring to standard definition frames, as defined in the GEM terminal specification.

N.1 Active Format Definition

The documentation for the method `getActiveFormatDefinition()` of the class `org.dvb.media.VideoFormatControl` specifies that the active_format field of the Active Format Descriptor shall be returned if it is present. GEM terminal specifications shall specify signalling that is used to determine this value. A GEM terminal specification may specify a different descriptor to be signalled. If a different descriptor is signalled, the method shall in all cases return the corresponding value as defined in TS 101 154 [2], annex B.

NOTE: The "public final static int" values specified in the class `VideoFormatControl` may be inlined at the time an application's Java source is compiled. Therefore, it is essential that the same value be used in all GEM terminal specifications, e.g. the same numeric value of DAR_16_9 (as specified by TS 101 154 [2]) will be returned, even if the underlying signalling used by the GEM terminal specification uses a different value to indicate the same thing.

N.1.1 MHP Signalling for Active Format Definition

GEM terminals obeying a GEM terminal specification that adopts the MHP definition of the functional equivalent named "active format descriptor" shall accept the Active Format Descriptor as specified by the documentation for the method `getActiveFormatDefinition()` of the class `org.dvb.media.VideoFormatControl` as specified in MHP [1], annex N.

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

GEM terminal specifications 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.service 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 the present document. GEM terminal specifications may make other filesystems available via this API (e.g. for files stored in local storage within the GEM terminal). These other file systems may subset the semantics of this package as follows:

- `DSMCCObject.abort()` - shall always throw a `NothingToAbortException;`
- `DSMCCObject.asynchronousLoad(AsynchronousLoadingEventListener)` - if the file exists, succeeds immediately (with `SuccessEvent` being generated) otherwise fails with `InvalidPathNameException;`
- `DSMCCObject.isObjectKindKnown()` - always returns `true;`
- `DSMCCObject.isStream()` - always returns `false;`
- `DSMCCObject.isStreamEvent()` - always returns `false;`
- `DSMCCObject.loadDirectoryEntry(AsynchronousLoadingEventListener)` - always succeeds immediately with `SuccessEvent` being generated;
- `DSMCCObject.prefetch(both signatures)` - always returns `false;`
- `DSMCCObject.setRetrievalMode(int)` - silently ignored;
- `DSMCCObject.synchronousLoad()` - if the file exists, succeeds immediately otherwise fails with `InvalidPathnameException.`

Additionally

- `ObjectChangeEvent` instances shall not be generated.
- The `DSMCCObject.unload()` method shall not be considered as removing stored files from where they are stored.
- Access to files shall not fail for reasons of a service domain not being in an attached state even though a stored filesystem cannot be represented by a DSMCC Service Domain.

NOTE 1: These semantics are identical to an agreed solution for MHP 1.1.2

NOTE 2: In MHP, the carousel from which an application is run is automatically mounted. Therefore, MHP applications that use org.dvb.dsmcc do not necessarily use the ServiceDomain class.

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 shall 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. 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.

NOTE: See also the semantic limitations on methods of this class under certain conditions specified in clause P.1.

P.2.1.1 DSMCCObject.getSigners()

GEM terminal specifications may be written such that some applications are trusted without being authenticated by the MHP mechanism. For these applications, a GEM terminal specification may be written such that it is a valid implementation option that this method returns an outer array of size zero when the DSMCCObject is loaded, if the file is not signed using the required MHP mechanism.

GEM terminal specifications may define a functionally equivalent mechanism for indicating the signers of files, as described in clause 12.2, "Authentication of applications." If such a mechanism is defined, the signers of a file shall be returned by this method.

NOTE: In the case where the MHP signing mechanism is not included in a GEM terminal specification, clause 12.2, "Authentication of applications" requires the presence of a functionally equivalent mechanism to indicate signers of files.

P.2.1.2 DSMCCObject.getSigners(boolean known_root)

NOTE: The semantics of the DSMCCObject.getSigners(boolean known_root) method are unchanged and require processing.

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.2.3.1 Lightweight binding of trigger API

GEM terminal specifications may define a lightweight binding to the trigger API, which consists of `DSMCCStreamEvent` and related classes. The lightweight binding is recommend where signalling is available that cannot reasonably support the full DSMCC trigger API. This may be in addition to or instead of a signalling binding to the full API, such as is provided by the DSMCC signalling.

For the lightweight binding, no signalling is required for the trigger object, described in clause B.4.1, "Trigger object." Instead, at a minimum a single DSMCCStreamEvent object named `"lightweight_triggers"` shall be available at the top level for appropriate instances of ServiceDomain.

NOTE 1: GEM terminal specification may specify signalling that allows the name of this object to be signalled, or to otherwise generalize the minimum requirements laid out here.

NOTE 2: It is recommended that all services that can carry triggers have a defined binding to the trigger API. Applications may subscribe to named trigger events (defined in clause B.4.2, "Trigger event") using this instance of `DSMCCStreamEvent`. The GEM terminal specification defines a binding to determine the source of the trigger events for this instance. See also the example in clause W.2

P.2.3.1.1 DSMCCStreamEvent.getEventList()

For the lightweight binding of the trigger API, this method may return null.

P.2.3.1.2 StreamEvent.getEventId()

For the lightweight binding of the trigger API, this method may always return -1 to indicate that there is no event ID.

P.2.3.1.3 DSMCCStreamEvent.unsubscribe(int, StreamEventListener)

For the lightweight binding of the trigger API, this method may always throw `UnknownEventException`.

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 to be present by the present document, however, signalling to support an NSAP address is not required. In GEM Terminal Specifications where such signalling is not defined, it is a valid implementation for this method to do nothing.

NOTE: The MHP definition of the functional equivalent named "Carousel" defines signalling for the NSAP address. See also clause 15.6.1.1.1, "NSAP Address".

P.2.5.2 ServiceDomain.attach

P.2.5.2.1 ServiceDomain.attach(Locator)

The locator parameter is to be interpreted as any locator that refers to a service domain. Locator formats are discussed in clause 6.3.3.

P.2.5.2.2 ServiceDomain.attach(Locator, int)

The locator parameter is to be interpreted as any locator that refers to a service. The integer is to be interpreted as a unique deification of a service domain within that service. Locator formats are discussed in clause 6.3.3.

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 an equivalent 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.

NOTE: As specified by javax.tv.locator.Locator.equals(), an equivalent locator has the same external form.

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 6 in clause 14.8, "Locators and content referencing." 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 6.

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.2.5.6 ServiceDomain.isNetworkConnectionAvailable()

The specification for this method is considered to be replaced by:

This method shall return true if and only if the source(s) of data for this filesystem are connected and physically

available to the terminal.

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. References to "NPT" and "eventNPT" refer to "timebase values" and "timebase_value", respectively.

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 S is included in the present document, with the following notes and modifications:

S.1 Additional database filter types

The methods getAppIDs(AppDatabaseFilter) and getAppAttributes(AppsDatabaseFilter) on the class AppsDatabase contains the following text:

> For implementations conforming to this version of the specification, only `CurrentServiceFilter` or `RunningApplicationsFilter` filters may return a non empty `Enumeration`. If the filter object is not an instance of `CurrentServiceFilter` or `RunningApplicationsFilter` or a subclass of either then, the method shall return an empty `Enumeration`.

If a GEM terminal specification defines additional filter types, then the requirement to return an empty `Enumeration` shall not apply for these filter types.

Annex T (normative): Permissions

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

NOTE: See clause 11.10, "Java permissions."

Annex U (normative): Extended graphics APIs

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

U.1 org.dvb.net.tuning.DvbNetworkInterfaceSIUtil

NOTE: As a consequence of clause 11.8.3, "Additional permissions classes," the packages `org.dvb.net.ca` and the class `org.dvb.net.tuning.DvbNetworkInterfaceSIUtil` are not required under certain circumstances.

U.2 org.dvb.ui.DVBGraphics.setXORMode()

The following behavior is specified in MHP [1] clause U in the documentation for the class `org.dvb.ui.DVBGrahpics`:

- Calling `setXORMode` on an instance of this class shall be equivalent to calling `setDVBComposite` with a special and implementation dependent `DVBAlphaComposite` object which implements the semantics specified for this method in the parent class.
- Calling `getDVBComposite` when `setXORMode` is the last DVBComposite set shall return this implementation dependent object. Conformant MHP applications shall not do anything with or to this object including calling any methods on it.

This behavior is not required for GEM terminals, and GEM applications shall not rely on this behavior.

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;
                case TYPE_UNKNOWN:
                case TYPE_OTHER:
                    success = setupOTHER(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.

W.3 Example of testing for optional APIs

```
    /**
     * This is an example of writing application code that uses
     * an optional API that may not be present on all
     * platforms.  In this case, the application tests for
     * the presence of the class org.dvb.net.rc.RCInterfaceManager.
     * If it is present, then the application has available to
     * it that API, and other APIs needed for the return channel.
     * <p>
     * This is the main Xlet class
     **/

    public class MyXlet implements javax.tv.xlet.Xlet {

        ...

        public RCFeatures rcFeatures = null;
        // This object is the gateway to all of the Xlet
        // code that relies on the return channel.  If
        // null, the Xlet is running on a box with no
        // return channel capability.

        public void initXlet(javax.tv.xlet.XletContext ctx) {
        ...

        try {
            Class c = Class.forName(
                            "org.dvb.net.rc.RCInterfaceManager");
            // We have the return channel API
            c = Class.forName("RCFeaturesImpl");
                        // An Xlet class
            rcFeatures = c.newInstance();
                        // Calls default constructor
            rcFeatures.init();
        } catch (ClassNotFoundException ex) {
            // No return channel, so we leave rcFeatures null.
        }
        }
    }

    /**
     * This interface is used by the Xlet to call into
```

```
 * code that uses the return channel in any way.  It
 * is essential that the only code in the Xlet that
 * uses classes not present on the EB profile be reached
 * via the class that implements this interface.  If an
 * Xlet directly references an API from some other place, then
 * the entire Xlet might fail to load on valid MHP
 * implementations.
 * <p>
 * In technical terms, when a class is loaded, a valid Java
 * implementation may load the transitive closure of all
 * statically-referenced classes, and fail to load the first
 * class if any of the other classes are not found.  Thus,
 * if an Xlet directly references an API that is not present,
 * the entire Xlet could fail to load on valid MHP
 * implementations.  As examples, static references can
 * be the types of data members or local variables in code
 * blocks, even if those code blocks are never executed.
 * <p>
 * To put the APIs that might not be present outside of the
 * transitive closure of statically referenced classes, we
 * introduce this interface, and dynamically load the single
 * class that implements it using Class.forName().  The class
 * that implements this interface can contain static references
 * to APIs that might not be present, and it can contain static
 * references to classes that reference such classes.
 *  <p>
 *  This interface can be thought of as a Facade.
 *  See the Facade pattern on page 185 of GoF
 *  ("Design Patterns" by Gamma et al, ISBN
 *  0-201-63361-2).
 **/
 public interface RCFeatures {

    /**
     * Set up the return channel, and get ready for the Xlet
     * to run.
     **/
    public void init();

    ... Here, there are declarations of all of the features
        of the Xlet that use the return channel.  They're
    probably quite high-level, such as "submit survey
    results."  There might be other methods, to manage
    the return channel during Xlet state transitions,
    such as to the paused or destroyed state  ...

 }


 /**
  * This class contains implementations of all features of the
  * xlet that depend on the presence of a return channel.  It
  * can safely reference classes that are not present in the
  * EB profile, and it can safely reference classes that
  * reference such classes.
  **/

public class RCFeaturesImpl implements RCFeatures {

    private org.dvb.net.rc.RCInterfaceManager mgr;

    public void init() {
    mgr = org.dvb.net.rc.RCInterfaceManager.getInstance();
    ...  use mgr to set up the return channel, as
                necessary ...
    }

    ...  Here, there are implementations of the features
         declared above ...

}
```

W.4 Example of lightweight trigger API

```
public class example implements StreamEventListener {
```

```
  public void example() {
    //  s is a ServiceDomain that carries a lightweight trigger binding.  An
    // interoperable GEM application might get the locator to the service domain
    // from a simple text file used for configuration to a certain platform.

    ServiceDomain s = ....

    DSMCCObject topLevelObject = s.getMountPount();
    DSMCCObject triggerObject = new DSMCCObject( topLevelObject,
        "lightweight_triggers");

    //  This assumes a lightweight trigger binding with names "a" to "z" hard-coded.  If
    //  the name of the trigger might vary based on platform or signalling, this name could
    // be parameterized through a text file, as with the ServiceDomain's locator.

    DSMCCStreamEvent eventSource = new DSMCCStream( triggerObject );
    eventSource.subscribe("a", this);
  }
  public void receiveStreamEvent( StreamEvent e )
  {
      //....
  }
}
```

Annex X (normative): Test support

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

NOTE: The introduction and the figure are specific to broadcast systems. It is not applicable to the packaged media target. Many of the descriptive elements in this annex are based on a broadcast system; the details for both broadcast and packaged media implementations may differ.

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

MHP [1], annex Y is included in the present document with the following modification:

The class description of `org.dvb.io.IxcRegistry` contains the following clause:

NOTE: ORGANIZATION ID values between 0x80000000 and 0xffffffff are known to create problems in some implementations and their use is strongly discouraged.

In the description of the method `lookup(XletContext, String)` the following note is added:

NOTE: ORGANIZATION ID values between 0x80000000 and 0xffffffff are known to create problems in some implementations and their use is strongly discouraged.

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
V1.2.1	May 2004	Publication
V1.3.1	October 2005	Publication
V1.4.1	May 2008	Publication