



Draft GLOBALLY EXECUTABLE MHP
Draft 1.0 for review
Document mug015r4 - DRAFT, DO NOT
REDISTRIBUTE
“ITU Scrubbed” version

6 June 2002

Reproduction of the document in whole or in part without prior permission of the DVB Project Office is forbidden.

Status of this Document

Revision	Date	Comment
0	04/04/2002	First edition of GEM 1. This has the status of Sun contribution.
1	10/04/2002	Added list of open actions, comments. Added at least a general discussion of the contents of every chapter, through the third appendix. This version is still a Sun contribution; it is the version being submitted for consideration at meeting 4 in Tokyo.
2	10/04/2002	Snapshot immediately after updates made during meeting 4
3	5/21/2002	Implementation of meeting 4 conclusions and fleshing out of document, including all chapters through 16 plus annex G.
"ITU Scrubbed"	06/06/2002	Removal of comments for ITU, at the request of Microsoft

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://www.etsi.org/legal/home.htm>).

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 the 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.

0 Introduction

0.1 Purpose

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

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

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

Despite these regional differences, it is desirable to be able to execute an interoperable MHP 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. This achieves binary interoperability of interactive content; the application code does not need to be modified when moving a service between differing network infrastructures. The service simply needs to be encoded according to the requirements of the underlying network.

This specification 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 must normatively reference the GEM specification in its entirety, and must fulfil the normative requirements of GEM.

0.2 Format

This specification 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 must be built or what network infrastructure must underlie the implementation; it is limited to specifying the behavior 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 behavior 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.

0.3 Inclusion of MHP features

0.3.1 Subsetting Prohibited

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

0.3.2 Supersetting Permitted

If a terminal specification based on GEM wishes to include APIs, signalling or behaviors defined in [MHP \[1\]](#) that are not required by GEM, it may do so by referring directly to [MHP \[1\]](#).

0.4 Addition of non-GEM Interfaces

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

In no case shall it be necessary for a GEM application to use such an extension interface in order to enable behavior that is normatively required by this specification.

0.4.1 DVB-J enumerations

A terminal specification based on GEM shall not add new values to an enumeration that is returned from a method defined by this specification.

NOTE: For example, the interface `org.dvb.net.rc.RCInterface` defined in [annexR](#), "(normative): [DVB-J Return Channel Connection Management API](#)" on page 57 introduces an enumeration that is returned by the method `getType()`, which includes the values `TYPE_CATV`, `TYPE_DECT`, etc. It

is not permissible to attempt to sub-divide one of these types by introducing new enumeration values in a different namespace. See also the example in annex W, "(informative): DVB-J examples" on page 62.

0.5 Application Areas

In this version of the GEM specification, the same application areas as [MHP \[1\]](#) section 0.2 are considered.

0.6 Profiles

As not all implementations supporting GEM will be able to support all application areas and as there is a further evolution expected over time, different profiles of GEM are considered. For this release of the GEM specification, profiles are mapped to the above mentioned application areas.

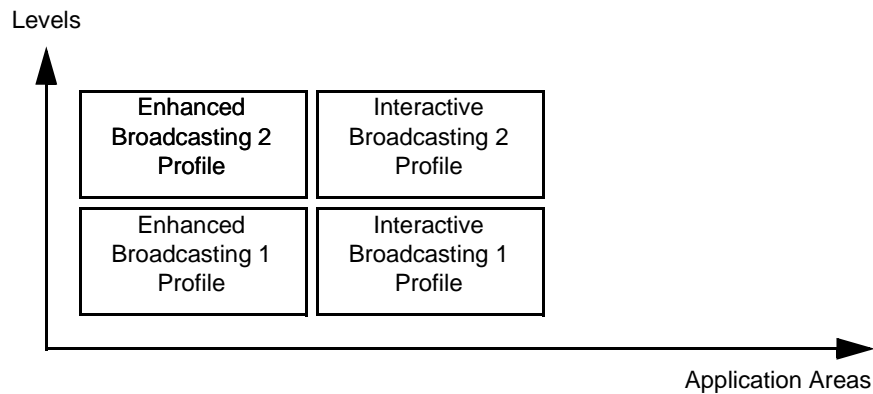


Figure 1 : Application areas and levels of profiles

Fig. 1 shows four example profiles, derived from two levels for each of the two application areas. The specific definition of the profiles and the particular backward and cross compatibility between profiles is provided in the detailed profile definition chapter of the GEM specification. The following initial definitions apply: <profile><n+1> shall be a strict superset of <profile><n>, and Interactive Broadcasting Profile 1 is defined as a strict superset of Enhanced Broadcasting Profile 1. Other dependencies are left to the detailed definition of future profiles.

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.

This specification 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 \[A\]](#), and [ARIB PE \[B\]](#). Implementers should consult the publisher of specifications which reference GEM regarding conformance.

NOTE: This specification 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 this specification is to minimize the number of divergences between MHP and terminal specifications based on GEM, wherever practical. [Divergence](#) is defined in section 3.1, "[Definitions](#)" on page 7. Where divergences are inescapable, this document serves as a place to document and control the permitted divergences, so that they will be predictable to terminal manufacturers, broadcasters, and application authors.

2 References

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

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

The following comments apply to particular sources of documents:

[1]	Where the reference is to an ISO specifications it is considered to be a "non-specific" reference additionally officially published amendments and corrigenda are considered to automatically update the referenced document.
[2]	Where an ISBN number is provided for a referenced document it is considered to be "specific reference".
[3]	References to RFCs are considered to be "specific references". An RFC being indicated obsoleted by another RFC is not considered significant.
[4]	URL references with note [4] are provided for convenience to access the document in electronic form.
[5]	URL references with note [5] are the normative method to access the reference
[6]	ETSI specifications are available from the ETSI server at: http://www.etsi.org . However, the ETSI server provides the current edition of the specification and in every case this specification makes "specific" references which in the future may not be the current reference.

Reference	Edition	Description	Note
[1] MHP	1.0.2	Digital Video Broadcasting (DVB), Multimedia Home Platform version 1.0.2	

3 Definitions and abbreviations

3.1 Definitions

3.1.1 Definitions from MHP

[MHP \[1\]](#) section 11.3 is considered to be included in this specification, with the following notes and modifications.

In the body of definitions only, the interpretations described [4.2, "Terminology in the MHP specification" on page 8](#) are to be applied.

3.1.2 Definitions Introduced by GEM

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

Binary Interoperability: In the context of GEM, the ability to run one application across a variety of terminals and middleware specifications without modification of files, except for files that need to be modified as a consequence of normal localisation. For example, class files may not be modified, but a text file holding locators or text messages might be replaced with a version tailored to a specific locale or network. A GEM application will be *binary interoperable* if it can in multiple environments with no change to program elements. e.g. for a DVB-J application, no change to the class files will be required, although some modification of non-program elements, such as user messages and locator external forms might be required.

Divergence: Everything that violates a spec assertion and/or a conformance clause. A divergence from the MHP specification is when a correctly written conformance test for an MHP specification assertion would fail.

GEM: Globally Executable MHP.

GEM Application: An application that is written to the interfaces defined in GEM that relies on only those interfaces and the semantic guarantees provided by the GEM specification. An interoperable GEM application will run on an MHP terminal, or on any terminal that complies to a terminal specification based on GEM.

GEM Terminal: A terminal or other device that can execute GEM applications. Examples of GEM terminals include an MHP terminal, and OCAP terminal or an ARIB AE terminal.

3.2 Abbreviations

3.2.1 Abbreviations from MHP

[MHP \[1\]](#) section 3.2 is considered to be included in this specification.

3.2.2 Abbreviations Introduced by GEM

For the purposes of the present document, the following abbreviations apply:

GEM	Globally Executable MHP
-----	-------------------------

4 Conventions

4.1 References within the MHP specification

[MHP \[1\]](#) contains numerous internal references. In certain cases, a section of the MHP specification that is referenced by GEM will refer to a section of the MHP specification that is not referenced by GEM, or to a section 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 don't apply or where they should be interpreted as referring to a corresponding section of GEM.

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

4.2 Terminology in the MHP specification

4.2.1 MHP

This document makes numerous references to [MHP \[1\]](#). When a section of the MHP specification is referenced from GEM, for the purposes of GEM references to MHP are to be interpreted to apply to GEM, and to terminal specifications based on GEM. Similarly, “MHP implementations” and “MHP terminal” are to be interpreted as “implementations of terminal specifications based on MHP,” etc. “MHP application” is to be interpreted as “GEM application.”

4.2.2 Resident Navigator

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

Downloaded or other resident applications that perform some of the policy decisions or functionality of the MHP navigator must implement a policy that is consistent with the requirements of this specification.

4.2.3 DVB Service

For the purposes of this document, references within [MHP \[1\]](#) to DVB services shall be interpreted as meaning any services which carry GEM applications.

4.3 Inclusion of sections of the MHP specification

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

5 Basic Architecture

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

6 Transport Protocols

6.1 Introduction

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

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

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

6.2 Broadcast Channel Protocols

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

Other protocols and their APIs are considered as extensions to this specification, see [H, "\(normative\): Extensions" on page 45](#).

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

The full details of APIs that provide DVB-J applications with access to broadcast protocols are in chapter 11, "[DVB-J Platform](#)" on page 19.

Except in the case of MPEG-2 sections (see [6.2.2, "MPEG-2 Sections" on page 10](#)), when a GEM application attempts to access conditional access scrambled data through one of these broadcast channel protocols, the terminal shall attempt to initiate descrambling of this data without the application needing to explicitly ask for it. Attempts to access conditional access scrambled data at the level of MPEG-2 sections shall not happen without the application explicitly asking for this.

6.2.1 MPEG-2 Transport Stream

[MHP \[1\]](#) section 6.2.1 is considered to be included in this specification.

6.2.2 MPEG-2 Sections

[MHP \[1\]](#) section 6.2.2 is considered to be included in this specification.

6.2.3 DSM-CC Private Data

[MHP \[1\]](#) section 6.2.3 is considered to be included in this specification.

Note: Use of this protocol is optional for terminal specifications based on GEM, however a functional equivalent must be specified.

6.2.4 DSM-CC Data Carousel

[MHP \[1\]](#) section 6.2.4 is considered to be included in this specification.

Note: Use of this protocol is not required for terminal specifications based on GEM, however a functional equivalent must be specified.

6.2.5 DSM-CC User-to-User Object Carousel.

[MHP \[1\]](#) section 6.2.5 is considered to be included in this specification, with the following notes and modifications. For this section, sub-sections are only included if this is explicitly indicated.

Note: Use of this protocol is not required for terminal specifications based on GEM, however a functional equivalent must be specified.

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

6.2.5.1 DVB-J class files

[MHP \[1\]](#) section 6.2.5.1 is considered to be included in this specification, 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 this section of [MHP \[1\]](#).

6.2.5.2 DVB-HTML document files

Void

6.2.5.3 Loss of Carousel Behaviour

[MHP \[1\]](#) section 6.2.5.3 is considered to be included in this specification, with the following notes and modifications.

The conditions for permanent loss of a carousel may be specified differently from MHP in terminal specifications based on GEM, therefore the reference to [MHP \[1\]](#) section B.2.11 need not necessarily apply. This specification does include [MHP \[1\]](#) section 9.1, thus the conditions for temporary disconnection and reconnection of a carousel as defined in [MHP \[1\]](#) section 9.1.5 do apply. Thus, the language in [MHP \[1\]](#) section 6.2.5.3 following the first paragraph to apply to this specification.

6.2.6 Protocol for delivery of IP multicast over the broadcast channel.

[MHP \[1\]](#) section 6.2.6 “DVB Multiprotocol Encapsulation” is considered to be included in this specification, with the following notes and modifications.

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

6.2.7 Internet Protocol (IP)

[MHP \[1\]](#) section 6.2.7 is considered to be included in this specification.

6.2.8 User Datagram Protocol (UDP)

[MHP \[1\]](#) section 6.2.8 is considered to be included in this specification.

6.2.9 DVB Service Information

[MHP \[1\]](#) section 6.2.9 is considered to be included in this specification.

Note: Use of this protocol is not required for terminal specifications based on GEM, however some form of service information must be defined. For DVB-J applications, the DVB SI APIs are not mandatory, but Java TV provides generic SI access APIs.

6.3 Interaction Channel Protocols

[MHP \[1\]](#) section 6.3 is considered to be included in this specification, with the following notes and modifications.

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

GEM terminals that support IP shall be compatible with Internet Protocol as defined in [MHP \[1\]](#) section 6.3.2.

GEM terminals that support TCP shall be compatible with Transmission Control Protocol as defined in [MHP \[1\]](#) section 6.3.3.

GEM terminals that support UDP shall be compatible with UDP as defined in [MHP \[1\]](#) section 6.3.9.

7 Content formats

7.1 Static Formats

[MHP \[1\]](#) section 7.1 is considered to be included in this specification.

7.2 Broadcast streaming formats

7.2.1 Audio

Some format for streaming audio must be specified in a terminal specification based on GEM.

7.2.2 Video

Some format for streaming video must be specified in a terminal specification based on GEM.

7.2.3 Subtitles

Support for subtitle signalling is optional in this specification.

Note: [MHP \[1\]](#) specifies one possible signalling scheme for subtitles.

Note: [OCAP \[A\]](#) 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 \[1\]](#) section 7.3 is considered to be included in this specification.

7.4 Downloadable Fonts

[MHP \[1\]](#) section 7.4 is considered to be included in this specification.

7.5 Colour Representation

[MHP \[1\]](#) section 7.5 is considered to be included in this specification.

7.6 MIME Types

[MHP \[1\]](#) section 11.5 is considered to be included in this specification, with the following notes and modifications.

Note: The entries for “image/dvb.subtitle”, “text/dvb.subtitle”, “text/dvb.teletext” and “multipart/dvb.service” refer to content types for which support is not required by this specification.

8 DVB-HTML

DVB-HTML is not a required or optional component in any profile defined in this specification, or in [MHP \[1\]](#). For this reason, this specification contains no definition of DVB-HTML.

9 Application model

9.1 Broadcast GEM Applications

[MHP \[1\]](#) section 9.1.1 is considered to be included in this specification, with the following notes and modifications.

In this section, the terms “AIT” and “application descriptor” are to be interpreted as referring to the application description defined in [10.4, "Application Description" on page 15](#). The term “DVB service” is to be interpreted as meaning “service.” Additionally, attention is drawn to the general rules in section [4.2, "Terminology in the MHP specification" on page 8](#).

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

In section 9.1.5, the reference to [MHP \[1\]](#) section 6.2.5.3 is to be interpreted as referring to [6.2.5.3, "Loss of Carousel Behaviour" on page 11](#). The language at the end of [MHP \[1\]](#) section 9.1.5 relating to the PMT information only applies to terminal specifications that feature this signalling.

9.2 DVB-J Model

[MHP \[1\]](#) section 9.2 is considered to be included in this specification, with the following notes and modifications.

The reference to the `application_control_code` parameter of the AIT in [MHP \[1\]](#) section 9.2.3.2 is to be interpreted as referring to the `application_control_code` defined in [10.4, "Application Description" on page 15](#).

9.3 DVB-HTML Model

Void

9.4 Inter application resource management

[MHP \[1\]](#) section 9.4 is considered to be included in this specification, 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 [10.4, "Application Description" on page 15](#).

Some downloaded resident applications specified as extensions to this specification may perform some of the functions of the MHP navigator, e.g. the monitor application defined [OCAP \[A\]](#). In this case, such downloaded software must implement a policy that is consistent with the requirements of this specification, e.g. [MHP \[1\]](#) section 9.4.

10 Application Signalling

10.1 Introduction

This section 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 \[1\]](#) gives an example 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 must be used or the packaging of applications; this is left for GEM-based specifications to define.

10.1.1 Summery of requirements on common signalling

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

- Some form of [Application Description \[10.4\]](#) with information sufficient to:
 - Identify the source of the application code and other assets
 - Identify the application's application ID and organisation 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 summarised as follows:

- A [DVB-J specific application description \[10.5\]](#) with information sufficient to:
 - signal parameters to the application
 - indicate the initial class of the application

10.2 Program Specific Information

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

- The [Application Description \[10.4\]](#) for each application in the service.
- The source of the application code and data.

10.3 Locators Within an Application Description

Some fields of the application description contain locators, e.g. locators to a directory containing certain kinds of files. These locators can be to any transport defined within a terminal specification based on GEM, 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 must 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 must 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 \[1\]](#) section 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 40 seconds.

Note: [MHP \[1\]](#) section 10.4.2 meets this requirement with a limit of 30 seconds.

10.4.2 Visibility of Application Description

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

10.4.3 Content of the Application Description

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

Table 1 : Application Description

Function	Type
<code>application_type</code>	enumeration
<code>organisation_id</code>	32 bit unsigned integer
<code>application_id</code>	16 bit unsigned integer
<code>application_control_code</code>	enumeration
<code>application_profiles</code>	see text
<code>version.major</code>	8 bit unsigned integer
<code>version.minor</code>	8 bit unsigned integer
<code>version.micro</code>	8 bit unsigned integer
<code>service_bound_flag</code>	boolean
<code>visibility</code>	enumeration
<code>application_priority</code>	8 bit unsigned integer
<code>application_name</code>	String
<code>application_icon_locator_count</code>	unsigned integer
for (<code>i=0</code> ; <code>i<N</code> ; <code>i++</code>) {	
<code>application_icon_locator</code>	Locator
<code>application_icon_flags</code>	16 bit unsigned integer
}	

application_type: Identifies the type of application. Currently, the only type of application that GEM specifies is DVB-J application, however signalling to identify at least 255 different application types in the future is recommended.

organisation_id: An `organisation_id`, as defined in [MHP](#) section 10.5.1 under `organisation_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](#) section 10.5.1 under `application_id`.

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

application_profiles: Information sufficient to derive the MHP profile or profiles on which this application could execute, as defined in [MHP](#) section 10.7.3. If a terminal specification based on GEM only features support of one MHP profile, than no explicit signalling of application profile is required.

version.major: The major sub-field of the profile version number, as defined in [MHP](#) section 10.7.3.

version.minor: The minor sub-field of the profile version number, as defined in [MHP](#) section 10.7.3.

version.micro: The micro sub-field of the profile version number, as defined in [MHP](#) section 10.7.3.

The four fields above indicate the minimum MHP profile on which an application will run. For example, an application that relies on the guarantees of GEM 1.0 would run on an appropriate profile of MHP 1.0.2. The underlying signalling of the application must 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](#) section 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](#) section 10.7.3.

application_priority: An application priority, as defined in [MHP](#) 10.7.3. Terminal specifications based on GEM must support at least 256 priority levels, with the semantics spelled out in MHP's definition of this flag.

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 language code with each application name. It shall be possible to signal any string that can be represented with UTF8.

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

application_icon_locator: Information sufficient to derive a locator to a directory containing application icons. The application icons shall be in files in the directory indicated by this locator, in the format specified in [MHP](#) section 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](#) section 10.7.4.

10.4.4 Applications from Previously Selected Services

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

10.5 DVB-J specific application description

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

10.5.1 Content of DVB-J application description

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

Table 2 : DVB-J Application Description

Function	Type
for (i=0; i<N; i++) { <code>dvbj_app_parameter</code> }	String
<code>base_directory</code>	Locator
for (i=0; i<N; i++) { <code>classpath_element</code> (optional) }	Locator
<code>initial_class_name</code>	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; cf. section 10.9.1.

initial_class_name: The fully-qualified name of the initial class of this application. This class must implement the Xlet interface. The signalling must support an whose UTF8 encoding is 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](#) section 10.9.2 defines the classpath_extension for this purpose. If support for this is included in a terminal specification, there may be restrictions placed on these locators, e.g. that they represent sub-directories of the [base_directory](#).

11 DVB-J Platform

11.1 The Virtual Machine

[MHP \[1\]](#) section 11.1 is considered to be included in this specification.

11.2 General Issues

[MHP \[1\]](#) section 11.2 is considered to be included in this specification.

11.3 Fundamental DVB-J APIs

[MHP \[1\]](#) section 11.3 is considered to be included in this specification, with the following notes and modifications.

Note: [MHP \[1\]](#) section 11.3.1.1 section g does apply to this specification. Thus, all terminal specifications based on GEM require support for the system property “dvb.persistent.root”.

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

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

This specification 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 [14.8, "Locators and content referencing" on page 30](#).

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: 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 concrete 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 \[1\]](#) section 11.4 is considered to be included in this specification, with the following notes and modifications.

Note: As a consequence of section [0.4 on page 4](#), the requirements of [MHP \[1\]](#) section 11.4.1.4 are required of all GEM-based terminal specifications; as a further consequence, it shall not be required for an application to call a method not required by this specification in order to obtain this behavior.

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

The last paragraph of [MHP \[1\]](#) section 11.4.2.2 places a requirement on the handling of `DVBLocators`. As `DVBLocator` is not required by this specification, this paragraph does not apply. Instead, this specification requires that any information in a locator referring to a specific program event is to be ignored by JMF players. See also [14.8, "Locators and content referencing" on page 30](#).

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

11.5 Data Access APIs

[MHP \[1\]](#) section 11.5 is considered to be included in this specification, with the following notes and modifications.

The reference to annex P in [MHP \[1\]](#) section 11.5.1 is to be read as referring to annex P of this specification.

The reference to annex R in [MHP \[1\]](#) section 11.5.5 is to be read as referring to annex R of this specification.

11.6 Service Information and Selection APIs

11.6.1 DVB Service Information API

The DVB specific SI API is **not** required in this specification. Thus, [MHP \[1\]](#) section 11.6.1 is not considered to be included in this specification.

11.6.2 Service Selection API

[MHP \[1\]](#) section 11.6.2 is considered to be included in this specification.

11.6.3 Tuning API

[MHP \[1\]](#) section 11.6.3 is considered to be included in this specification, with the following notes and modifications.

The reference to the DvbLocator class does not apply to this specification. The reference to MHP section 11.7.6 is to be read as referring to section [11.7.6, "Content Referencing" on page 22](#) of this document.

11.6.4 Conditional Access API

[MHP \[1\]](#) section 11.6.4 is considered to be included in this specification.

11.6.5 Protocol Independent SI API

[MHP \[1\]](#) section 11.6.5 is considered to be included in this specification, with the following notes and modifications.

The mapping of the protocol independent SI API onto the underlying SI protocol is not defined in this specification. Thus, the reference to [MHP \[1\]](#) annex O does not apply. However, terminal specification based on GEM must provide a mapping of the protocol independent SI API onto their SI signalling.

11.7 Common Infrastructure APIs

11.7.1 APIs to support DVB-J application lifecycle

[MHP \[1\]](#) section 11.7.1 is considered to be included in this specification, with the following notes and modifications.

Note: The xlet properties “dvb.org.id”, “dvb.app.id” and “dvb.caller.parameters” must be supported.

In [MHP \[1\]](#) section 11.7.1.1, the reference to the DVB-J application descriptor is to be interpreted as referring to ["Content of DVB-J application description" on page 17](#). The text requiring that a specific text encoding be used does not apply to this specification.

11.7.2 Application discovery and launching APIs

This API is formed of the `org.dvb.application` package defined in ["\(normative\): Application Listing and Launching" on page 58](#).

Note: This is the same API as in [MHP \[1\]](#).

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

Table 3 : Application attribute properties

Property name (note 1)	Return
dvb.j.location.base	Returns String containing base_directory from the DVB-J Application Description .
dvb.j.location.cpath.extension	Returns String[] derived from classpath_element from the DVB-J Application Description with each array entry corresponding to a pathname entry as defined for classpath_element .
NOTE 1: Property names beginning "dvb." are reserved for future use.	

Note: The property `dvb.transport.oc.component.tag` is not required by this document.

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

Table 4 : Information source for methods on AppAttributes

Method	Information source
<code>getName()</code>	One of the names that can be found in the application_name of the Application Description .
<code>getName(String ISO639code)</code>	A name of the application_name of the Application Description corresponding to the specified language, if available.
<code>getNames()</code>	All of the names for the application which can be found in the application_name of the Application Description and their ISO 639 language code.
<code>getProfiles()</code>	The set of profiles indicated in the application_profiles of the Application Description .
<code>getPriority()</code>	The value indicated for the application_priority of the Application Description .
<code>getVersions(String profile)</code>	The values version.major , version.minor and version.micro for the specified profile from the Application Description .
<code>getIsServiceBound()</code>	True if the service_bound_flag of the Application Description indicates true. Otherwise false.
<code>isStartable()</code>	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 terminal specification based on GEM, if the are supported.
<code>getIdentifier()</code>	The organisation_id and application_id of the Application Description .
<code>getServiceLocator()</code>	If remote applications are supported, the locator for a remote application shall encapsulate the values found in the appropriate signalling in the terminal specification.
<code>getLocator()</code>	The application_icon_locator of the Application Description .
<code>getIconFlags()</code>	The application_icon_flags of the Application Description .

11.7.3 Inter-Application communication API

[MHP \[1\]](#) section 11.7.3 is considered to be included in this specification.

11.7.4 Basic MPEG Concepts

[MHP \[1\]](#) section 11.7.4 is considered to be included in this specification, with the following notes and modifications.

The classes `DvbElementaryStream`, `DvbService`, and `DvbTransportStream` are not required by this specification. 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 this specification.

11.7.5 Resource Notification

[MHP \[1\]](#) section 11.7.5 is considered to be included in this specification.

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 \[1\]](#) section 11.7.6. The `DAVIC` class `DvbLocator` is not required by this specification.

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 [14.8, "Locators and content referencing" on page 30](#).

In this specification, 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 \[1\]](#) section 11.7.7 is considered to be included in this specification.

11.8 Security

[MHP \[1\]](#) section 11.8 is considered to be included in this specification.

11.9 Other APIs

11.9.1 Timer Support

[MHP \[1\]](#) section 11.9.1 is considered to be included in this specification.

Note: The minimum repeat interval of 40ms in MHP was motivated by a standard definition frame rate of 25Hz, 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 \[1\]](#) section 11.9.2 is considered to be included in this specification, with the following notes and modifications.

11.9.3 Profile and version properties

[MHP \[1\]](#) section 11.9.3 is considered to be included in this specification, with the following notes and modifications.

All of the system properties defined in this section are required by this specification. 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 chapter 15 of this specification. The properties referring to version numbers are to be interpreted as referring to the corresponding *MHP* version number.

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

11.10 Java permissions

[MHP \[1\]](#) section 11.10 is considered to be included in this specification, 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 [12.1.3, "Extensions to MHP application signing framework" on page 26](#), terminal specifications based on GEM may introduce other mechanisms for establishing that an application is trusted. Thus, in this section 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.

[MHP \[1\]](#) section 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.

11.11 Content referencing

The following mapping shall be used between the types of locator defined in [\[\[xref to equivalent of table 64 on page 213 of MHP\]\]](#) and the DVB-J methods defined in this chapter. It lists the Java methods and constructors which 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 these locators shall as described in [\[\[same xref\]\]](#) 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 section.

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 this specification. Other forms of locator from among those defined in this specification 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. This specification does not prevent methods accepting other forms of locator which are not defined in this specification.

11.11.1 Transport stream

[MHP \[1\]](#) section 11.11.1 is considered to be included in this specification, with the following notes and modifications.

The term “DVB locators” is considered to refer to all valid locators as described in [\[\[xref to table 64\]\]](#).

11.11.2 Network

[MHP \[1\]](#) section 11.11.2 is considered to be included in this specification, with the following notes and modifications.

The term “DVB network” is to be interpreted as referring to a valid network, as described in [\[\[xref to table 64\]\]](#).

11.11.3 Bouquet

[MHP \[1\]](#) section 11.11.3 is considered to be included in this specification, with the following notes and modifications.

The term “DVB bouquet” is to be interpreted as referring to a valid bouquet, as described in [\[\[xref to table 64\]\]](#).

11.11.4 Service

11.11.4.1 MPEG/GEM specific service

[MHP \[1\]](#) section 11.11.4.1 is considered to be included in this specification, 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 this specification:

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

11.11.4.2 Generic Service

[MHP \[1\]](#) section 11.11.4.1 is considered to be included in this specification, 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 \[1\]](#) section 11.11.5 is considered to be included in this specification, 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 this specification:

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

11.11.6 MPEG elementary stream

[MHP \[1\]](#) section 11.11.6 is considered to be included in this specification, with the following notes and modifications.

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

The following methods are *not* required by this specification:

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

11.11.7 File

[MHP \[1\]](#) section 11.11.7 is considered to be included in this specification, with the following notes and modifications.

The note about “instances of ‘dvb:’ locator including dvb_abs_path” shall be interpreted as meaning [[whatever the equivalent is, with a reference to the table 64 replacement]].

11.11.8 Directory

[MHP \[1\]](#) section 11.11.8 is considered to be included in this specification, with the following notes and modifications.

The phrase “‘dvb:’ locator” shall be interpreted as meaning GEM locator.

11.11.9 Drip feed decoder

[MHP \[1\]](#) section 11.11.9 is considered to be included in this specification.

11.11.10 Irrelevant

[MHP \[1\]](#) section 11.11.10 is considered to be included in this specification.

11.11.11 Methods working on many Locator types

The following methods used in this specification 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 [14.8, "Locators and content referencing" on page 30](#)) 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, no other `SIElements` exist

11.11.12 Support for the HTTP protocol in DVB-J

[MHP \[1\]](#) section 11.11.12 is considered to be included in this specification.

12 Security

12.1 Introduction

This section 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 \[1\]](#) section 12.1.1 is considered to be included in this specification.

12.1.2 Overview of return channel security

[MHP \[1\]](#) section 12.1.2 is considered to be included in this specification.

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 might involve some form of codesigning, but they are not required to.

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

- Require that trusted applications be identified with an `application_id` from the signed applications range, as described in [12.1.1](#).
- Support the permission request file as specified in [MHP \[1\]](#) section 12.6.2.
- Refuse to grant permissions outside of the set granted to unsigned applications in MHP, unless those permissions are explicitly requested in the signalling.

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

12.1.4 Extensions to MHP permission request file

Terminal specifications based on GEM might introduce new permissions that are not granted by default. Because of this, it might be desirable to define syntax to allow applications to request these additional permissions. Of course, any application that requires these additional permissions would not be guaranteed to run on all GEM-based terminal specifications.

As detailed in [12.6 "Security policy for applications"](#), the exact syntax of the MHP permission request file must be supported. Terminal specifications based on GEM may additionally extend the MHP security framework by supporting a different permission request file syntax. This syntax must:

- Use a `PublicLiteral` value in a namespace other than DVB's
- Use a `SystemLiteral` whose domain is not DVB's

If a non-DVB permission request file supports the same tags as the MHP permission request file, the names of any additional tags must be chosen so that they will not conflict with future versions of the MHP specification. For example, tags introduced by ARIB could contain the prefix "arib." in the tag name.

Note: One way to extend the MHP security model with additional permissions would be to allow applications to be packaged with two permission request files. In this scheme, the MHP permission request file would be used to request permissions defined in MHP, and the additional permission request file would only be used for non-MHP permissions.

12.2 Authentication of applications

[MHP \[1\]](#) section 12.2 is considered to be included in this specification.

12.3 Message transport

[MHP \[1\]](#) section 12.3 is considered to be included in this specification.

12.4 Detail of application authentication messages

[MHP \[1\]](#) section 12.4 is considered to be included in this specification.

Note: The exact file names, locations and syntaxes described in this section must be supported. This includes the requirement in section 12.4.3.1 that the last certificate in a CertificateFile be the root certificate.

12.5 Profile of X.509 certificates for authentication of applications

[MHP \[1\]](#) section 12.5 is considered to be included in this specification.

12.6 Security policy for applications

[MHP \[1\]](#) section 12.6 is considered to be included in this specification, with the following notes and modifications.

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

Note: The policy for granting of permissions outlined in [MHP \[1\]](#) section 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 \[A\]](#) defines a “monitor application” that makes policy decisions. In cases such as this, the downloaded element is required to implement a policy consistent with this specification.

Note: The exact syntax of the permission request file specified in [MHP \[1\]](#) section 12.6.2 is required to be supported. Because a reference to the DTD is a part of this file, it cannot be extended by adding tag definitions. Possible mechanisms for extending the MHP security model with additional permissions are outlined in [12.1.4 "Extensions to MHP permission request file"](#).

Table 54 in [MHP \[1\]](#) section 12.6.2.4 contains a reference to the name `initial_class_byte`. For this specification, this is to be interpreted as referring to `initial_class_name` from [table 2 on page 17](#).

[MHP \[1\]](#) section 12.6.2.9 refers to the AIT of a service. For this specification, this is to be interpreted as referring to the application description, as described in [10.4 "Application Description"](#).

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 section 12.6.1 relating to the return channel permission and return channel connections where it is not necessary to dial a phone, e.g. cable modems.

12.7 Example of creating an application that can be authenticated

MHP [1] section 12.7 is considered to be included in this specification.

12.8 GEM/MHP certification procedures

Certification procedures are outside the scope of this document.

12.9 Certificate management

MHP [1] section 12.9 is considered to be included in this specification.

12.10 Security on the return channel

MHP [1] section 12.10 is considered to be included in this specification.

12.11 the internet profile of X.509 (informative)

MHP [1] section 12.11 is considered to be included in this specification.

12.12 Platform minima

MHP [1] section 12.12 is considered to be included in this specification.

13 Graphics reference model

[MHP \[1\]](#) chapter 13 is considered to be included in this specification, with the following notes and modifications.

13.1 Supported graphics resolutions

[MHP \[1\]](#) chapter 13 contains references to the platform minima in section G, e.g. in sections 13.2.1.3 and 13.3.6.1. For this document, these references are to be interpreted as referring to [\(normative\): Minimum Platform Capabilities \[G\]](#) in this document.

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

13.2 Broadcast streaming formats

[MHP \[1\]](#) section 13.4.1 mandates background players for the broadcast streaming formats. This specification does not mandate a particular broadcast streaming format, e.g. Standard Definition 25Hz MPEG Video is not required by this specification. A terminal specification based on GEM must include some mechanism for delivering audio and video programming, however. For these formats, background JMF players must be created. Thus, the last paragraph of section 13.4.1 applies to this player.

13.3 Subtitles

Signalling to support subtitles is not required by this specification. [MHP \[1\]](#) section 13.5 only applies to terminal specifications based on GEM for which subtitling is available. If subtitling signalling is available, the presentation of subtitles must follow the model specified in [MHP \[1\]](#) section 13.5.

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

14 System integration aspects

14.1 Namespace mapping

This specification does not mandate any particular format for locators. Note, however, that [14.8, "Locators and content referencing" on page 30](#) requires that terminal specifications based on this document define some text representation for certain entities.

14.2 Reserved names

[MHP \[1\]](#) section 14.2 is considered to be included in this specification.

14.3 XML notation

[MHP \[1\]](#) section 14.3 is considered to be included in this specification, with the following notes and modifications.

These XML notation rules only apply to XML file formats defined in this specification, 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. This specification relaxes this prohibition: Terminal specifications based on this specification may extend the allowed XML notation by permitting other character encodings. If no encoding is specified, however, the default shall be UTF-8.

14.4 Network signalling

This specification does not mandate specific network signalling, nor error behavior when incorrectly formatted data is received.

14.5 Text encoding of application identifiers

[MHP \[1\]](#) section 14.5 is considered to be included in this specification, with the following notes and modifications.

The references to `organisation_id` and `application_id` are to be interpreted as referring to the versions of [organisation_id](#) and [application_id](#) from this specification.

14.6 Reserved names for persistent storage

[MHP \[1\]](#) section 14.6 is considered to be included in this specification.

14.7 Files and file names

[MHP \[1\]](#) section 14.8 is considered to be included in this specification, 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 [14.8 "Locators and content referencing"](#).

14.8 Locators and content referencing

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

Note: [11.7.6, "Content Referencing" on page 22](#) describes how this text representation can be used in DVB-J applications.

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

Table 5 : Addressable entities, locators and their text representation

Entity	Text Representation
Transport stream	locator text representation must be defined
Network	No standardised text representation required
Bouquet	No standardised text representation required
Service	locator text representation must be defined
Service Domain	locator text representation must be defined.
Program Event	locator text representation must be defined
MPEG Elementary Stream	DVB locator including "dwb_service_component" element
File	"file:", "http:" and "https:" URLs, as referred to in MHP [1] section 14.8 locator text representation must be defined for files located within a Service Domain .
Directory	"file:", "http:" and "https:" URLs, as referred to in MHP [1] section 14.8 locator text representation must be defined for directories located within a Service Domain .
Drip feed decoder	"dripfeed://"

14.9 Service identification

Java TV features 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 both kinds of locators.

14.10 CA system

[MHP \[1\]](#) section 14.6 is considered to be included in this specification.

15 Detailed platform profile definitions

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

Table 6 : Detailed platform profile definitions (Sheet 1 of 2)

Area	Specification	Enhanced Broadcast Profile 1	Interactive Broadcast Profile 1	Internet Access Profile 1
Static formats				
Bitmap pictures	MHP [1] section 7.1.1.3, "PNG" + 15.1, "PNG - restrictions" on page 33	M	M	
	MHP [1] section 7.1.1.3, "PNG" without restrictions	-	-	
	MHP [1] section 7.1.1.4, "GIF"	-	-	
	MHP [1] section 7.1.2 "MPEG-2 I-Frames"	M	M	
	MHP [1] section 7.1.1.2 "JPEG" + 15.3, "JPEG - restrictions" on page 34	M	-	
	MHP [1] section 7.1.1.2 "JPEG" without restrictions	-	M	
Audio clips	MHP [1] section 7.1.4 "Monomedia format for audio clips"	M	M	
Video drips	MHP [1] section 7.1.3 "MPEG-2 Video 'drips'"	M	M	
Text encoding	MHP [1] section 7.1.5 "Monomedia format for text"	M	M	
Broadcast streaming formats				
Video	7.2.2, "Video" on page 12	M	M	
Audio	7.2.1, "Audio" on page 12	M	M	
Subtitles	7.2.3, "Subtitles" on page 12	-	-	
Fonts				
Built in	Character set see annex E, "(normative): Character set" on page 42, Metrics see annex D, "(normative): Text presentation" on page 41 Face: UK RNIB "Tiresias"	M	M	
Downloadable	7.4, "Downloadable Fonts" on page 12	M	M	
Broadcast channel protocols				
	6.2.2, "MPEG-2 Sections" on page 10	M	M	
	6.2.5, "DSM-CC User-to-User Object Carousel." on page 10	-	-	
	IP Multicast stack based on: 6.2.6, "Protocol for delivery of IP multicast over the broadcast channel." on page 11, 6.2.7, "Internet Protocol (IP)" on page 11, 6.2.8, "User Datagram Protocol (UDP)" on page 11	O	Ro	
Interaction channel protocols				
TCP/IP	MHP [1] section 6.3.3 "Transmission Control Protocol (TCP)" MHP [1] section 6.3.2 "Internet Protocol (IP)"	-	M	
UDP/IP	6.2.8, "User Datagram Protocol (UDP)" on page 11, MHP [1] section 6.3.2 "Internet Protocol (IP)"	-	M	
HTTP	MHP [1] section 6.3.7.1 "HTTP 1.1"	-	O	
DVB-J				
Core	11.3, "Fundamental DVB-J APIs" on page 19	M	M	
Presentation	MHP [1] section 11.4.1, "Graphical User Interface API"	M (note 1)	M (note 1)	
	MHP [1] section 11.4.2, "Streamed Media API"	M	M	

Table 6 : Detailed platform profile definitions (Sheet 2 of 2)

Area	Specification	Enhanced Broadcast Profile 1	Interactive Broadcast Profile 1	Internet Access Profile 1
Data Access	MHP [1] section 11.5.1, "Broadcast Transport Protocol Access API" as modified by 11.5, "Data Access APIs" on page 20	M	M	
	MHP [1] section 11.5.2, "Support for Multicast IP over the Broadcast Channel"	O	Ro	
	MHP [1] section 11.5.3, "Support for IP over the Return Channel"	-	M	
	MHP [1] section 11.5.4, "MPEG-2 Section Filter API"	M	M	
	MHP [1] section 11.5.5, "Mid-Level communications API" as modified by 11.5, "Data Access APIs" on page 20	-	M	
	MHP [1] section 11.5.6, "Persistent Storage API"	M	M	
Service Information & Selection	11.6.1, "DVB Service Information API" on page 20	-	-	
	11.6.2, "Service Selection API" on page 20	M	M	
	11.6.3, "Tuning API" on page 20	M	M	
	11.6.4, "Conditional Access API" on page 20	M	M	
	11.6.5, "Protocol Independent SI API" on page 20	M	M	
Common Infrastructure	11.7.1, "APIs to support DVB-J application lifecycle" on page 20	M	M	
	11.7.2, "Application discovery and launching APIs" on page 20	M	M	
	11.7.3, "Inter-Application communication API" on page 21	M	M	
	11.7.4, "Basic MPEG Concepts" on page 21	M	M	
	11.7.5, "Resource Notification" on page 22	M	M	
	11.7.6, "Content Referencing" on page 22	M	M	
	11.7.7, "Common Error Reporting" on page 22	M	M	
Security	MHP [1] section 11.8.1 "Basic Security"	M	M	
	MHP [1] section 11.8.2 "APIs to Support TLS / SSL Over the Return Channel"	-	M	
	MHP [1] section 11.8.3 "Additional permissions classes"	M	M	
Others	11.9.1, "Timer Support" on page 22	M	M	
	11.9.2, "User Settings and Preferences API" on page 22	M	M	
	11.9.3, "Profile and version properties" on page 22	M	M	
NOTE 1: The javax.tv.graphics.TVContainer.getRootContainer method shall return an instance of org.havi.ui.HScene or null except when called by an embedded Xlet where the behaviour is defined in 11.13.2, "Embedded Xlets in a DVB-HTML Page" on page 283.				

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

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

15.1 PNG - restrictions

MHP [1] section 11.9.1 is considered to be included in this specification.

15.2 Minimum media formats supported by DVB-J APIs

MHP [1] section 11.9.1 is considered to be included in this specification, with the following notes and modifications.

Support for subtitles is optional.

15.3 JPEG - restrictions

MHP [1] section 15.3 is considered to be included in this specification.

15.4 Locale support

MHP [1] section 15.3 is considered to be included in this specification.

Note: Terminal specifications may, of course, guarantee support for locales in addition to UK English, however, support for the UK English local is required by this specification.

15.5 Video raster format dependencies

This section addresses the aspects of this specification that vary as a consequence of the video raster format.

15.5.1 Standard Definition (PAL/SECAM or NTSC resolution)

15.5.1.1 Logical pixel resolution

The logical pixel resolution shall be 72 dots per inch.

16 Registry of Constants

16.1 System constants

[MHP \[1\]](#) section 16.1 is considered to be included in this specification.

16.2 DVB-J constants

[MHP \[1\]](#) section 16.2 is considered to be included in this specification, with the following notes and modifications.

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

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

[MHP \[1\]](#) annex A is considered to be included in this specification.

Annex B (normative): Broadcast Filesystem and Trigger Transport

This specification does not specify a signalling format for broadcast file systems or for trigger (event) delivery. It does, however, require that terminal specifications based on GEM provide a mechanism for delivery of filesystems and triggers.

NOTE: [MHP \[1\]](#) Annex C defines a profile of DSMCC object carousels which fulfills the requirements of this annex.

B.1 Service Domain

Terminal specifications based on GEM must 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 sections.

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.

A service domain provides a “mount point” for a file system. Once an application attaches to a service domain, a mapping from locators to files within the carousel is established. To read any object in a filesystem, an interoperable GEM application must first mount a service domain, then navigate the filesystem to the desired object. The text form of a locator to a file system object other than a service domain is implementation dependant.

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, "\(normative\): Broadcast Transport Protocol Access"](#) on page 53.

B.2 Filesystem Requirements

B.2.1 Static Requirements

Terminal specifications based on GEM must include a mechanism for delivering a hierarchical file system within a service domain. It must be possible to construct a locator that refers to files and directories in this hierarchy. The file system delivery mechanism must 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 1 : Filesystem Signalling Requirements

Area	Minimum Requirement
Characters Allowed in File Names	The ASCII character ‘a’..‘z’, ‘A’..‘Z’, ‘0’..‘9’, ‘-’, ‘.’ and ‘_’. After the first character of a file name, ‘ ’ (the space character) is also permitted.
Maximum length of file name	200 characters
Number of entries per directory	10,000
Maximum Directory Nesting	20 levels
Maximum File Length	$2^{31}-1$ bytes
Caching	Optional, but if supported, signalling to disable caching shall be provided.

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.

Note: The following requirements are modeled on DSMCC BIOP::StreamMessage.

Stream descriptions shall be identified with a special file sent in the hierarichal filesystem described in section B.2, "Filesystem Requirements" on page 37. This file shall contain information sufficient to derive the following.:

Table 2 : Stream Description

Function	Type
<code>npt_source</code>	Reference (see text)
<code>stream_locator</code>	Locator external form
<code>duration</code>	32 bit unsigned integer
<code>audio_stream</code>	Reference (see text)
<code>data_stream</code>	Reference (see text)
<code>data_stream</code>	Reference (see text)
<code>is_mpeg_program</code>	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.

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

audio_stream: A reference to an audio stream.

video_stream: A reference to a video stream.

data_stream: A reference to a stream containing data that is neither audio nor video.

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

B.4 Trigger Signalling

There shall be a mechanism for sending triggers to an application.

NOTE: The following requiements are modeled on DSMCC stream events

B.4.1 Trigger Object

Triggers shall be identified with a special file sent in the hierarichal filesystem described in section B.2, "Filesystem Requirements" on page 37. This file shall contain information sufficient to derive all of the contents of a [Stream Description](#), plus the following.:

Table 3 : Trigger Object

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

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

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

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

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

B.4.2 Trigger Event

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

Table 4 : Trigger Event

Function	Type
<code>event_id</code>	14 bit unsigned integer
<code>is_do_it_now</code>	flag
<code>mpeg_npt</code>	32 bit unsigned integer
<code>payload</code>	byte array

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

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

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

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

B.4.2.1 Extrapolation of NPT Values

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

NOTE: This corresponds to the requirements in NPT signalling spelled out in [MHP \[1\]](#) section B.2.4.4, “Timebases.”

B.4.2.2 Monitoring of trigger events

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

NOTE: This corresponds to the requirements spelled out in [MHP \[1\]](#) section B.2.4.5, “Monitoring Stream Events.” MHP requires that scheduled stream event descriptors be *broadcast* at least 5 seconds 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 considered to be included in this specification.

C.2 Additional Informative References

	Reference	Edition	Description
[A]	OCAP	Rev.11	OCAP
[B]	ARIB PE		

Annex D (normative): Text presentation

[MHP \[1\]](#) Annex D is considered to be included in this specification, with the following notes and modifications.

Table D.3 in [MHP \[1\]](#) section D.3.4.2 is an informative table giving the pixel aspect ratio for regions where standard definition television is 720x576 pixels. This table may not apply in all regions, e.g. regions with NTSC standard definition.

Annex E (normative): Character set

[MHP \[1\]](#) Annex E is considered to be included in this specification.

Annex F (informative): Authoring & Implementation Guidelines

[MHP \[1\]](#) Annex F is considered to be included in this specification.

Annex G (normative): Minimum Platform Capabilities

G.1 Graphics

[MHP \[1\]](#) section G.1 is considered to be included in this specification, with the following notes and modifications.

The last bullet point of [MHP \[1\]](#) section G.1.1 discusses a resolution of 720x576. This is to be interpreted as meaning the appropriate resolution with the terminal in standard definition mode. An aspect ratio of 4:3 is required by this specification, but support for a 16:9 aspect ratio is optional.

[MHP \[1\]](#) section G.1.2 lists DFB_PROCESSING_LB_16_9 and DFC_PROCESSING_PAN_SCAN as mandatory in MHP. In this specification, support for these is optional.

G.2 Audio

[MHP \[1\]](#) section G.2 is considered to be included in this specification.

G.3 Video

[MHP \[1\]](#) section G.3 is considered to be included in this specification.

G.4 Resident fonts and text rendering

[MHP \[1\]](#) section G.4 is considered to be included in this specification.

G.5 Input events

[MHP \[1\]](#) section G.5 is considered to be included in this specification.

Some downloaded resident applications specified as extensions to this specification may perform some of the functions of the MHP navigator, e.g. the monitor application defined [OCAP \[A\]](#). In this case, such downloaded software must implement a policy that is consistent with the requirements of this specification, e.g. [MHP \[1\]](#) section G.5.

G.6 Memory

[MHP \[1\]](#) section G.6 is considered to be included in this specification

G.7 Other resources

[MHP \[1\]](#) section G.7 is considered to be included in this specification, 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 this specification; however, attention is drawn to the requirement on detecting application description changes in [10.4.1](#), "Application Description transmission and monitoring" on page 16.

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

Annex H (normative): Extensions

[MHP \[1\]](#) Annex H is considered to be included in this specification.

Annex I (normative): DVB-J fundamental classes

[MHP \[1\]](#) Annex I is considered to be included in this specification.

Annex J (normative): DVB-J event API

[MHP \[1\]](#) Annex J is considered to be included in this specification.

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

[MHP \[1\]](#) Annex K is considered to be included in this specification.

Annex L (normative): User Settings and Preferences API

[MHP \[1\]](#) Annex L is considered to be included in this specification, 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 this specification 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

The API defined in [MHP \[1\]](#) Annex M is *not* required by this specification.

Annex N (normative): Streamed Media API Extensions

[MHP \[1\]](#) N is considered to be included in this specification, with the following notes and modifications.

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

NOTE: In regions with NTSC standard definition, this is 720x480.

References to ETR 154 are to be read as referring to the standard definition format specification required by the terminal specification.

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

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

Annex P (normative): Broadcast Transport Protocol Access

P.1 Overview

The API defined in this section allows DVB-J applications direct access to information broadcast according to Annex B, "(normative): Broadcast Filesystem and Trigger Transport" on page 37. Of course, terminal specifications based on GEM may make other filesystems available via this API.

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

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

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

The following, briefly explains the functionalities offered by this API

The ServiceDomain class enables attaching to a [Service Domain](#).

When attached to a Service Domain, objects are available representing the types File, Directory, [Stream Description](#), [Trigger Object](#) and [Trigger Event](#).

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

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

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

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

The StreamEvent class represents an abstraction of the real event that is generated, i.e. the [Trigger Event](#), which enables the broadcaster to synchronize the application with the stream. This class enables the access to the content of an event, as described in section B.4.2, "[Trigger Event](#)" on page 39.

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

P.2 The org.dvb.dsmcc package

This package is derived from the org.dvb.dsmcc package as defined in [MHP \[1\]](#) annex P. A small number of the MHP methods are not required in terminal specifications based on GEM. Additionally, in GEM these methods are bound to the more abstract signalling requirements of Annex B on page 37.

The description of each class from the org.dvb.dsmcc package defined in [MHP \[1\]](#) annex P is considered to be included in this specification, 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 this specification.

P.2.1 DSMCCObject

The first sentence of the class description should be interpreted as meaning that this class represents objects in a [Service Domain](#).

P.2.2 DSMCCStream

References to BIOP::Stream message are to be interpreted as meaning the stream description as defined in Annex B.3, "[Stream Description](#)" on page 37. References to the BIOP::StreamEvent message are to be interpreted as meaning the stream even description defined in Annex B.4.1, "[Trigger Object](#)" on page 38. 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](#) reference refers to an audio stream.

P.2.2.2 isData() method

This shall return true if the [data_stream](#) reference refers to a data stream.

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](#) reference refers to a video stream.

P.2.3 DSMCCStreamEvent

References to the BIOP::StreamEvent message are to be interpreted as meaning the stream even description defined in Annex B.4.1, "[Trigger Object](#)" on page 38. 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 B.4.1, "[Trigger Object](#)" on page 38.

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 Annex B.1, "[Service Domain](#)" on page 37.

Throughout this class, references to "service gateway" or "service domain" are to be interpreted as referring to service domain, as described in Annex B.1, "[Service Domain](#)" on page 37.

P.2.5.1 ServiceDomain.attach(byte[])

Signalling to support the ServiceDomain.attach(byte[]) method is not required by this specification. In terminal specifications where no such signalling is defined, the behavior of invoking this method may be undefined.

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

The locator parameter is to be interpreted as any locator that refers to a service domain. Locator formats are discussed in section 14.8, "[Locators and content referencing](#)" on page 30.

P.2.5.3 ServiceDomain.getLocator()

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

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

P.2.5.4 ServiceDomain.getNSAPAddress()

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

P.2.5.5 ServiceDomain.getURL(Locator)

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

Returns a URL corresponding to a locator referring to a file or a directory, as specified in Table 5, "[Addressable entities, locators and their text representation](#)" on page 31. If the service domain corresponding to the locator is attached and the file or directory referenced in the locator exists then an instance of java.net.URL is returned which can be used to reference this object.

Parameters:

l - a locator referring to a file or directory, as specified in Table 5, "[Addressable entities, locators and their text representation](#)" on page 31.

Returns:

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

Throws:

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

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

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

P.2.6 ServiceXFRErrorEvent

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

P.2.7 ServiceXFRException

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

P.2.8 ServiceXFRReference

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

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 section B.4.2, "[Trigger Event](#)" on page 39. References to the event data refer to the [payload](#) defined in that section.

Annex Q (normative): Datagram Socket Buffer Control

[MHP \[1\]](#) Annex Q is considered to be included in this specification.

Annex R (normative): DVB-J Return Channel Connection Management API

[MHP \[1\]](#) Annex R is considered to be included in this specification.

Annex S (normative): Application Listing and Launching

[MHP \[1\]](#) Annex L is considered to be included in this specification.

Annex T (normative): Permissions

[MHP \[1\]](#) Annex T is considered to be included in this specification.

Annex U (normative): Extended graphics APIs

[MHP \[1\]](#) Annex U is considered to be included in this specification.

Annex V : Void

Annex W (informative): DVB-J examples

W.1 DVB-J examples from MHP

MHP [1] annex W is considered to be included in this specification.

[m6 add:]

W.2 Example of enumeration extension

To illustrate the importance of the requirement in section 0.4.1, "DVB-J enumerations" on page 4, consider an application that is written to the GEM specification which wishes to query the type of return channel connection, and react accordingly. Such code might be written in the following manner:

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

public class AppRCTester {

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

        .... definition of methods setupCATV et al.
    }
}
```

If it were permissible for a terminal specification based on GEM 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 behavior of GEM would not work if the enumeration value were returned by a method in a class with a constructor that is accessible to applications, because it would be impossible to mandate that all instances conforming to certain criteria implement an additional interface. In this case, other extension mechanisms would need to be employed.

Annex X (normative): Test support

[MHP \[1\]](#) Annex X is considered to be included in this specification.

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

[MHP \[1\]](#) Annex Y is considered to be included in this specification.

Annex Z (informative): Services, Service Contexts and Applications in an MHP Environment

[MHP \[1\]](#) Annex Z is considered to be included in this specification, with the following notes and modifications.

This informative section includes references to some signalling details not required by this specification. 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 this specification.