Exhibit 1024



Digital Video Broadcasting (DVB); Globally Executable MHP (GEM) Specification 1.3 (including OTT and hybrid broadcast/broadband)

DVB Document A153

March 2011

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238


DVB
Digital Video
Broadcasting

Contents

Intellectual Property Rights 29
Foreword 29
Introduction 29
Purpose 29
Aim 30
1 Scope 31
2 References 32
2.1 Normative references 32
2.2 Informative references 36
3 Definitions and abbreviations 38
3.1 Definitions 38
3.2 Abbreviations 42
4 General considerations and conventions 44
4.1 General considerations 44
4.1.1 Purpose 44
4.1.2 Format 44
4.1.3 Inclusion of GEM features 44
4.1.3.1 Subsetting prohibited 44
4.1.3.2 Supersetting permitted 45
4.1.4 Addition of non-GEM interfaces 45
4.1.4.1 DVB-J enumerations 45
4.1.4.2 Competitive APIs 45
4.1.4.2.1 Illustration of Complementary Functional equivalents 45
4.1.4.2.2 Illustration of Competitive APIs 46
4.1.5 Application areas 46
4.1.5.1 Broadcast target 47
4.1.5.2 Packaged Media target 47
4.1.5.3 IPTV target 47
4.1.5.4 OTT target 47
4.1.5.5. Typical Hybrid profiles 48
4.1.6 Profiles 48
4.1.7 Full conformance with the present document 48
4.2 Conventions 49
4.2.1 Void 50
4.2.2 Void 50
4.2.3 Void 50
4.2.4 Conventions within the present document 50
4.2.4.1 GEM 50
4.2.4.2 Resident navigator 50
4.2.4.3 DVB service 50
4.2.5 References to OCAP 50
5 Basic architecture (informative) 52
5.1 Context 52
5.2 Architecture 53
5.2.1 Resources 53
5.2.2 System software 53
5.2.2.1 Application Manager 53
5.2.3 Application 54
5.3 Interfaces Between a GEM Application and the GEM Terminal 54
5.4 Plug-ins 56
5.4.1 Security Model 57
6 Transport protocols 58

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

6.1 Introduction 58
6.2 Broadcast channel protocols 58
6.2.1 MPEG-2 transport stream 59
6.2.2 MPEG-2 sections 59
6.2.3 DSM-CC private data 59
6.2.4 DSM-CC data carousel 59
6.2.5 Object carousel 59
6.2.5.1 Void 59
6.2.5.2 Void 59
6.2.5.3 Loss of carousel behaviour 59
6.2.6 Protocol for delivery of IP multicast over the broadcast channel 60
6.2.7 Internet Protocol (IP) 60
6.2.8 User Datagram Protocol (UDP) 60
6.2.9 Service information 60
6.2.10 IP signalling 61
6.3 Interaction channel protocols 61
6.3.1 Network Dependent Protocols 61
6.3.2 Internet Protocol (IP) 62
6.3.3 Transmission Control Protocol (TCP) 62
6.3.4 UNO-RPC 62
6.3.5 UNO-CDR 62
6.3.6 DSM-CC User to User 62
6.3.7 Hypertext Transfer Protocol (HTTP) 62
6.3.7.1 HTTP 1.1 62
6.3.7.2 GEM profile of HTTP 1.0 62
6.3.7.2.1 HTTP 1.0 persistent connections 62
6.3.7.2.2 The Keep-Alive Header 63
6.3.7.2.3 GEM and proxies 63
6.3.7.2.4 Version compatibility 63
6.3.7.3 HTTPS 63
6.3.8 User Datagram Protocol (UDP) 63
6.3.9 DNS 63
6.3.10 Additional Transport Protocols 63
6.4 Transport protocols for application loading over the interaction channel 64
6.4.1 File system implemented only by the interaction channel 64
6.4.1.1 File system logical structure 64
6.4.1.2 File transfer 65
6.4.1.3 Class encoding 66
6.4.1.4 Directory listing in this file system 66
6.4.2 Hybrid between broadcast stream and interaction channel 66
6.4.2.1 File transfer 66
6.4.2.1.1 Broadcast file delivery 66
6.4.2.1.2 Interaction channel delivery 66
6.4.2.1.3 HTTPProfileBody 66
6.5 IPTV protocols 67
6.5.1 Transport protocols 67
6.5.1.1 Service Discovery and Selection 67
6.5.1.2 Broadband Content Guide 67
6.5.1.3 Real Time Protocol (RTP) 67
6.5.1.4 Real Time Streaming Protocol (RTSP) 68
6.5.1.5 Internet Group Management Protocol (IGMP) 68
6.5.2 Service information and metadata protocols 68
6.5.2.1 IP service discovery 68
6.5.2.2 Broadband content guide 68
6.6 OTT Protocols 68
6.6.1 Protocols for streaming 68
6.6.1.1 Adaptive Streaming 68
6.6.2 Protocols for download 69

7 Content formats 70
7.1 Static formats 70
7.1.1 Bitmap image formats 70

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

7.1.1.1 Image encoding restrictions 70
7.1.1.2 JPEG 70
7.1.1.3 PNG 70
7.1.1.4 GIF 70
7.1.2 MPEG-2 I-Frames 70
7.1.3 MPEG-2 Video "drips" 71
7.1.4 Monomedia format for audio clips 72
7.1.5 Monomedia format for text 72
7.1.5.1 Built-in character set 72
7.2 Media streaming formats 73
7.2.1 Audio 73
7.2.2 Video 73
7.2.3 Subtitles 73
7.2.3.1 DVB Subtitles 73
7.2.3.2 Teletext 73
7.2.4 Containers 74
7.2.5 Streaming Manifest 74
7.3 Resident fonts 74
7.4 Downloadable fonts 74
7.4.1 PFR 75
7.4.2 OpenType 76
7.5 Colour representation 76
7.5.1 Background (informative) 76
7.5.2 Specification 77
7.5.2.1 The sRGB Reference Viewing Environment 77
7.5.2.2 Colourimetric Definitions and Encodings 77
7.6 MIME types 79
7.6.1 Rationale 79

8 Void 80

9 Application model 81
9.1 Service-bound GEM applications 81
9.1.1 Basic lifecycle control 81
9.1.2 Starting applications 81
9.1.3 Support for execution of multiple simultaneous applications 82
9.1.4 Stopping applications 82
9.1.4.1 A new service being selected replacing a previously selected one 82
9.1.4.2 The stopping of an application by another application 82
9.1.4.3 Changes in the application signalling to request a particular application be stopped 82
9.1.4.4 Stopping by the GEM terminal due to a shortage of resources 82
9.1.5 Persistence of Applications Across Service Boundaries 83
9.1.6 Management of autostarting 83
9.1.7 Tuning without service selection 84
9.1.8 GEM Applications and Service Selection 84
9.1.9 Cached applications 84
9.1.9.1 Version management 85
9.1.9.2 Proactive caching 85
9.2 DVB-J Model 85
9.2.1 Starting DVB-J Applications 86
9.2.2 Stopping a DVB-J Application 86
9.2.3 DVB-J Application Lifecycle 86
9.2.3.1 Introduction 86
9.2.3.2 Lifecycle state machine for DVB-J application instances 86
9.2.4 Xlet API 89
9.2.4.1 Xlet State Change Semantics 89
9.2.4.2 Xlet state change requests 89
9.2.5 Multiple application environment support 90
9.2.5.1 Control of DVB-J applications by other DVB-J applications 90
9.2.5.2 Input Focus management 90
9.2.5.3 Other resources management 90
9.2.5.4 VM implementation 91

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

9.3 Void 91
9.4 Inter-application resource management 91
9.4.1 Application instances running in the same service context 91
9.4.2 Application instances not running in the same service context 92
9.5 Void 92
9.6 Services and applications not related to conventional services 92
9.6.1 Applications loaded from the interaction channel 92
9.6.2 Stored services 92
9.6.3 DVB-J Model 94
9.6.4 Common behaviour 94
9.7 Lifecycle of internet access applications 95
9.7.1 General issues 95
9.7.2 Starting internet access applications from GEM applications 95
9.7.3 Selecting DVB services from internet access applications 95
9.8 Plug-ins 96
9.9 Stored and cached applications 96
9.9.1 Storing files 96
9.9.2 Version management 97
9.9.3 Removing stored applications 97
9.9.4 Interrupted downloads 97
9.9.5 Dynamic behaviour 97
9.10 Lifecycle interactions between GEM and resident applications 98
9.11 Providers 98
9.11.1 Introduction (informative) 98
9.11.2 Lifecycle of xlet bound providers 99
9.11.3 Lifecycle of system bound providers 99
9.12 Impact of graphics constraints on the application model 99
9.12.1 Impact on generic applications 99
9.12.2 Impact on DVB-J applications 100
9.13 Unbound Applications 101
9.13.1 Introduction to unbound applications (informative) 101
9.13.1.1 Scope 101
9.13.1.2 Divergences from OCAP Solution 101
9.13.1.3 Overview 102
9.13.2 Service model 102
9.13.3 Application lifecycle 102
9.13.4 Initialization of GEM Environment 102

10 Application signalling 103
10.1 Introduction 103
10.1.1 Summary of requirements on common signalling 103
10.1.2 Summary of additional signalling for DVB-J applications 103
10.2 Program specific information 103
10.3 Locators within an Application Description 103
10.4 Application Description 104
10.4.1 Application Description transmission and monitoring 104
10.4.2 Visibility of Application Description and tuning 104
10.4.3 Content of the Application Description 104
10.4.3.1 DVB-J application control codes 107
10.4.3.2 Application icons descriptor 107
10.4.3.3 Graphics constraints descriptor 109
10.4.4 Applications from previously selected services 110
10.4.5 AIT File 110
10.4.9.1 Syntax 110
10.4.9.2 Syntactic restrictions 110
10.4.9.2.1 Transport protocols 110
10.4.9.3 Semantics 111
10.4.9.4 MIME type 111
10.5 DVB-J specific Application Description 111
10.5.1 General 111
10.5.2 Content of DVB-J Application Description 112
10.6 Constant Values 113

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

10.7 Plug-in signalling 113
10.7.1 Native signalling scenario 113
10.7.2 GEM signalling scenario 113
10.7.3 Delegated application descriptor 114
10.7.4 Plug-in descriptor 114
10.8 Stored Applications 115
10.8.1 Use of stored application signalling 115
10.8.1.1 Stored broadcast service related applications 115
10.8.1.2 Stored stand-alone applications 115
10.8.2 Application storage descriptor 116
10.8.3 Application Description File 117
10.8.3.1 Description 117
10.8.3.2 Application Description File name and location 118
10.8.3.3 Syntax 118
10.8.3.4 Semantics 118
10.9 Signalling for providers 119
10.10 Signalling for IPTV 119
10.10.1 Service bound application signalling 119
10.10.2 XAIT 120

11 DVB-J platform 121
11.1 The virtual machine 121
11.2 General issues 121
11.2.1 Basic Considerations 121
11.2.2 Approach to Subsetting 121
11.2.3 Class Loading 122
11.2.3.1 Fundamental principles 122
11.2.3.2 Class loading and providers 122
11.2.4 Unloading 122
11.2.5 Event listeners 122
11.2.6 Event model in DAVIC APIs 122
11.2.7 Event model in DAVIC and DVB APIs 122
11.2.8 Tuning as a side-effect 123
11.2.9 Intra application media resource management 123
11.2.10 Application thread priority 123
11.2.11 Text Encodings 123
11.2.11.1 Text encoding in Service Information 123
11.3 Fundamental DVB-J APIs 124
11.3.1 Java platform APIs 124
11.3.1.1 java.lang package 124
11.3.1.2 java.void 125
11.3.1.3 Void 125
11.3.1.4 java.io, javax.microedition.io 125
11.3.1.5 java.net 125
11.3.2 GEM platform APIs 126
11.3.2.1 org.dvb.lang 126
11.3.2.2 org.dvb.event 126
11.3.2.2.1 Generic description 126
11.3.2.2.2 Additional semantics for org.dvb.event 127
11.3.3 Java TV 127
11.4 Presentation APIs 127
11.4.1 Graphical User Interface API 127
11.4.1.1 The Core GUI API 127
11.4.1.2 TV user interface 128
11.4.1.3 Extended graphics 130
11.4.1.4 Television viewing mode 130
11.4.1.5 Font bindings 131
11.4.1.5.1 PFR0 131
11.4.1.5.2 OpenType 132
11.4.2 Streamed Media API 133
11.4.2.1 Framework of solution 133
11.4.2.2 Clarifications 133

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.4.2.3 Default media player behaviour 134
11.4.2.4 Required controls for video drips 134
11.4.2.5 Extensions to the Framework 134
11.4.2.5.1 DVB specified extensions 134
11.4.2.5.2 Extensions in org.davic 134
11.4.2.5.3 Extensions in javax.tv 135
11.4.2.5.4 Required controls for broadcast profiles and packaged media profiles 136
11.4.2.5.5 Clarifications 137
11.4.2.5.6 Component-based JMF players 137
11.4.2.5.7 Streaming Monitoring API 138
11.4.2.5.8 Media Stream Synchronization API 139
11.4.2.5.8.1 API behavior in border cases 139
11.4.2.5.8.2 Establishing a Master/Slave Relationship 139
11.4.2.5.8.3 Adding a Slave 140
11.4.2.6.8.4 Removing a Slave 140
11.4.2.5.8.5 Starting a Master Player 140
11.4.2.5.8.6 Setting the Media Time and Rate of a Master Player 141
11.4.2.5.8.7 Loss of Synchronization 141
11.4.2.5.8.8 Event Handling 141
11.4.2.6 Restrictions on the Framework for Broadcast 142
11.4.2.7 Intersection Between MediaSelectControl and SubtitlingLanguageControl/ AudioLanguageControl 143
11.4.2.8 Intersection between Streamed Media API and TV User Interface API 143
11.4.2.8.1 Basic Principles 143
11.4.2.8.2 TV Behaviour Control 144
11.4.2.8.3 Application Behaviour Control 144
11.4.2.8.4 Dynamic Behaviour 144
11.4.2.8.5 Resource Management Details 145
11.4.2.9 Integration with providers 145
11.4.2.10 Additional and modified semantics for IPTV 145
11.4.2.11 Time-setting operations for OTT 145
11.5 Data access APIs 146
11.5.1 Broadcast Transport Protocol Access API 146
11.5.1.1 Constraints on the java.io.File methods for broadcast carousels 146
11.5.1.2 Methods dealing with write access 147
11.5.1.3 Behaviour following loss of a broadcast carousel 147
11.5.2 Support for Multicast IP over the Broadcast Channel 148
11.5.3 Support for IP over the Return Channel 148
11.5.4 MPEG-2 Section Filter API 148
11.5.5 Mid-Level Communications API 148
11.5.6 Persistent Storage API 149
11.5.7 File Storage Device Access 151
11.5.7.1 Basic Specification 151
11.5.7.2 DVB specific modifications 151
11.6 Service information and selection APIs 151
11.6.1 Signalling-specific service information API 151
11.6.2 Service selection API 151
11.6.3 Tuning API 154
11.6.3.1 Generic description 154
11.6.3.2 Tuning in IPTV 154
11.6.3.3 Tuning in OTT 154
11.6.4 Conditional access API 154
11.6.5 Protocol independent SI API 155
11.6.5.1 Generic description 155
11.6.5.2 Transport independent and dependent services 155
11.6.5.3 Modified Semantics of Existing APIs 156
11.6.5.4 Extensions 156
11.6.6 Service discovery and selection for IPTV 156
11.6.7 Integration between protocol independent SI API and TV-Anytime 156
11.7 Common infrastructure APIs 156
11.7.1 APIs to support DVB-J application lifecycle 156
11.7.1.1 Xlet properties 157
11.7.1.2 Actions for DVB-J applications to perform in their destroy method 157

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.7.2 Application discovery and launching APIs 158
11.7.3 Inter-application communication API 159
11.7.4 Basic MPEG concepts 160
11.7.5 Resource notification 160
11.7.6 Content referencing 160
11.7.7 Common error reporting 161
11.7.8 Plug-in APIs 161
11.7.9 Provider API 162
11.7.9.1 API framework 162
11.7.9.2 SelectionProvider 162
11.7.9.3 SI providers 163
11.7.9.4 InteractionChannelTransportProvider 164
11.7.10 Content referencing for IPTV 164
11.7.11 TV-Anytime content referencing and metadata 164
11.7.12 Content referencing for OTT 165
11.8 Security 165
11.8.1 Basic Security 165
11.8.2 APIs for return channel security 165
11.8.3 Additional permissions classes 166
11.8.4 General Security Issues 166
11.8.5 Cryptographic API 166
11.8.6 DVB Extensions for Cryptography 166
11.8.6.1 Introduction (informative) 166
11.8.6.1.1 The org.dvb.security package 167
11.8.6.1.2 The org.dvb.auth.callback package 167
11.8.6.1.3 The org.dvb.net.ssl package 167
11.8.6.1.4 The org.dvb.security.pkcs11 package 167
11.8.6.2 Specification 167
11.9 Other APIs 168
11.9.1 Timer support 168
11.9.2 User settings and preferences API 168
11.9.3 Profile and version properties 168
11.9.3.1 Information on options 169
11.9.4 Non-CA smart card API 170
11.9.5 XML parsing API 171
11.9.5.1 SAX 171
11.9.5.2 JDOM 171
11.9.6 GEM terminal hardware API 171
11.9.7 Content Download API 171
11.10 Java permissions 171
11.10.1 Permissions for unsigned applications 171
11.10.1.1 java.awt.AWTPermission 171
11.10.1.2 java.net.SocketPermission 171
11.10.1.3 java.util.PropertyPermission 171
11.10.1.4 java.lang.RuntimePermission 172
11.10.1.5 java.io.SerializablePermission 172
11.10.1.6 java.io.FilePermission 172
11.10.1.7 javax.tv.media.MediaSelectPermission 172
11.10.1.8 javax.tv.service.ReadPermission 172
11.10.1.9 javax.tv.service.selection.ServiceContextPermission 172
11.10.1.10 java.util.Locale.setDefault 172
11.10.1.11 Applications signalled in AIT File 172
11.10.1.12 javax.microedition.xlet.ixc.IxcPermission 172
11.10.1.13 MonitorAppPermission and ServiceTypePermission 173
11.10.2 Additional Permissions for signed applications 173
11.10.2.1 java.util.PropertyPermission 173
11.10.2.2 java.io.FilePermission 173
11.10.2.3 org.dvb.net.ca.CAPermission 173
11.10.2.4 org.dvb.application.AppsControlPermission 174
11.10.2.5 org.dvb.net.rc.RCPermission 174
11.10.2.6 org.dvb.net.tuning.TunerPermission 174
11.10.2.7 javax.tv.service.selection.SelectPermission 174

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.10.2.8 org.dvb.user.UserPreferencePermission 174
11.10.2.9 java.net.SocketPermission 174
11.10.2.10 org.dvb.media.DripFeedPermission 175
11.10.2.11 org.dvb.application.storage.ApplicationStoragePermission 175
11.10.2.12 javax.microedition.apdu.APDUPermission 175
11.10.2.13 ServiceContextPermission 175
11.10.2.14 javax.microedition.xlet.ixc.IxcPermission 175
11.10.2.15 org.dvb.spi.ProviderPermission 175
11.10.2.16 Permissions for Unbound and Privileged Applications 175
11.11 Content referencing 176
11.11.1 Transport stream 176
11.11.2 Network 177
11.11.3 Void 177
11.11.4 Service 177
11.11.4.1 MPEG/GEM specific service 177
11.11.4.2 Generic service 178
11.11.4.3 Stored services 179
11.11.4.4 Content referencing for IPTV 179
11.11.4.5 Content referencing for OTT services 179
11.11.5 Program event 179
11.11.6 MPEG elementary stream 179
11.11.7 File 180
11.11.8 Directory 181
11.11.9 Drip feed decoder 181
11.11.10 Void 181
11.11.11 Methods working on many locator types 181
11.11.12 Support for the HTTP Protocol in DVB-J 182
11.11.13 GEM Applications 182
11.12 Stand-alone Applications 182
11.12.1 Common behavior 182
11.12.2 Stored services 183
11.12.2.1 Stored application APIs 183
11.12.2.2 Modified behaviour of GEM 1.0 APIs 183
11.12.2.3 Permissions 184
11.12.2.3.1 FilePermission 184
11.12.2.4 Stored application management API 184
11.13 Void 184
11.14 Internet Access 184
11.14.1 Internet client control APIs 184
11.14.2 Internet applet support 185
11.14.2.1 HTML tags 185
11.14.2.2 Java Platform 185
11.14.2.3 Void 185
11.14.2.4 Void 185
11.14.2.5 Security 185
11.15 APIs defined in OCAP 185
11.15.1 Introduction (informative) 185
11.15.2 OCAP Annex G - the org.ocap.application package 187
11.15.3 OCAP Annex P - the org.ocap.service package 187
11.15.4 OCAP Annex Q - the org.ocap.system package 188
11.15.5 OCAP annex O - the org.ocap package 188
11.15.6 OCAP annex U - the org.ocap system.event 188

12 Security 189
12.1 Introduction 189
12.1.1 Overview of the security framework for applications 189
12.1.2 Overview of return channel security 189
12.1.3 Establishing trusted applications 190
12.2 Authentication of applications 190
12.2.1 Overview of authentication messages 190
12.2.1.1 Hash codes 190
12.2.1.2 Signatures 190

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12.2.1.3 Certificates 191
12.2.1.4 Authentication of hierarchical file systems 191
12.3 Message transport 192
12.4 Detail of application authentication messages 192
12.4.1 HashFile 192
12.4.1.1 Description 192
12.4.1.2 HashFile location and naming conventions 193
12.4.1.3 Digest value computation rules 193
12.4.1.3.1 Example 194
12.4.1.4 Warning concerning grouping of objects under a single digest (informative) 194
12.4.1.5 Special authentication rules 194
12.4.2 SignatureFile 195
12.4.2.1 Description 195
12.4.2.2 SignatureFile location and naming conventions 195
12.4.2.3 Supported algorithms 196
12.4.2.4 Signature computation rules. 196
12.4.2.5 Authentication rules 196
12.4.3 CertificateFile 196
12.4.3.1 Description 196
12.4.3.2 ASN.1 encoding 197
12.4.3.3 Supported algorithms 197
12.4.3.4 Name matching 197
12.4.3.5 CertificateFile location and naming conventions 197
12.4.3.6 Authentication rules 197
12.4.4 Integration 197
12.5 Profile of X.509 certificates for authentication of applications 198
12.5.1 signatureAlgorithm 198
12.5.1.1 MD5 with RSA 198
12.5.1.2 SHA-1 with RSA 199
12.5.1.3 parameters 199
12.5.2 signatureValue 199
12.5.3 version 199
12.5.4 issuer 199
12.5.4.1 minimum requirement 199
12.5.4.2 certificate authority responsibility 199
12.5.5 validity 199
12.5.6 subject 199
12.5.7 SubjectPublic Key Info 200
12.5.7.1 rsaEncryption 200
12.5.7.2 subjectPublicKey 200
12.5.8 Unique Identifiers 201
12.5.9 Extensions 201
12.6 Security policy for applications 202
12.6.1 General principles 202
12.6.2 Permission request file 203
12.6.2.0 General 203
12.6.2.1 File encoding 204
12.6.2.1.1 XML 204
12.6.2.1.2 MHP/GEM 1.0 204
12.6.2.1.3 MHP/GEM 1.1 205
12.6.2.1.4 MHP/GEM 1.2 207
12.6.2.1.5 Number representation 209
12.6.2.2 File integrity 209
12.6.2.3 Example 209
12.6.2.4 Permission request file name and location 209
12.6.2.5 Permission Request file 210
12.6.2.5.1 Minimum permissions 210
12.6.2.5.2 Syntax and semantics 210
12.6.2.5.3 Defaults 210
12.6.2.6 Credentials 210
12.6.2.7 File Access 212
12.6.2.7.1 Unsigned applications 212

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 12 of 861

12.6.2.7.2 Policy for signed applications 212
12.6.2.7.3 Permission request syntax 213
12.6.2.8 CA API 213
12.6.2.8.0 GEM Introduction 213
12.6.2.8.1 Unsigned applications 213
12.6.2.8.2 Signed applications 214
12.6.2.8.3 Conditional Access Permission syntax 214
12.6.2.9 Application lifecycle control policy 214
12.6.2.9.1 Unsigned applications 214
12.6.2.9.2 Default policy for signed applications 215
12.6.2.9.3 Syntax 215
12.6.2.10 Return channel access policy 215
12.6.2.10.1 Unsigned applications 215
12.6.2.10.2 Signed applications 215
12.6.2.10.3 Return channel permission syntax 215
12.6.2.11 Tuning access policy 215
12.6.2.11.1 Unsigned applications 216
12.6.2.11.2 Signed applications 216
12.6.2.11.3 Tuner Permission syntax 216
12.6.2.12 Service selection policy 216
12.6.2.12.1 Unsigned applications 216
12.6.2.12.2 Signed applications 216
12.6.2.12.3 Service Selection Permission 216
12.6.2.13 Media API access policy 216
12.6.2.14 Inter-application communication policy 217
12.6.2.14.1 Unsigned applications 217
12.6.2.14.2 Signed applications 217
12.6.2.15 User Setting and Preferences access policy 217
12.6.2.15.1 Unsigned applications 217
12.6.2.15.2 Signed applications 217
12.6.2.15.3 Permission syntax 217
12.6.2.16 Network permissions 217
12.6.2.16.1 Unsigned applications 217
12.6.2.16.2 Signed applications 217
12.6.2.16.3 Permission syntax 217
12.6.2.17 Dripfeed permissions 218
12.6.2.17.1 Unsigned applications 218
12.6.2.17.2 Default policy for signed applications 218
12.6.2.17.3 Permission request syntax 218
12.6.2.18 Privileged Runtime Code Extension Permission 218
12.6.2.18.1 Unsigned Applications 218
12.6.2.18.2 Signed applications with no permission request file 218
12.6.2.18.3 Signed applications with a permission request file 218
12.6.2.18.4 Permission request syntax 218
12.6.2.19 Application storage 218
12.6.2.19.1 Unsigned applications 218
12.6.2.19.2 Default policy for signed applications 219
12.6.2.19.3 Permission request syntax 219
12.6.2.20 Non-CA smart card access 219
12.6.2.20.1 Unsigned applications 219
12.6.2.20.2 Default policy for signed applications 219
12.6.2.20.3 Permission request syntax 220
12.6.2.21 Provider Management 220
12.6.2.21.1 Unsigned applications 220
12.6.2.21.2 Signed applications 220
12.6.2.21.3 Permission request syntax 220
12.6.2.22 Service type selection policy 220
12.6.2.22.1 Unsigned applications 220
12.6.2.22.2 Signed applications 220
12.6.2.22.3 Permission request file syntax 220
12.6.2.23 Privileged application access 221
12.6.2.23.1 Unsigned applications 221

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12.6.2.23.2 Signed applications 221
12.6.2.23.3 Permission request file syntax 221
12.7 Example of creating an application that can be authenticated 221
12.7.1 Scenario Example 221
12.7.2 Hashes and signature computations: 222
12.7.2.1 Computation of the hashes of the root/Xlet1/classes/subclasses directory 222
12.7.2.2 Computation of the hashes of the of root/Xlet1/classes directory 222
12.7.2.3 Computation of the hashes of the of root/Xlet1 directory 223
12.7.2.4 Computation of the signature 224
12.8 Void 224
12.9 Certificate management 224
12.9.1 Certificate Revocation Lists 224
12.9.1.1 Introduction (informative) 224
12.9.1.2 Distribution of CRLs (informative) 224
12.9.1.2.1 Distribution via return channel 224
12.9.1.2.2 Distribution via MPEG stream 225
12.9.1.3 CRL retention 225
12.9.1.3.1 Requirement 225
12.9.1.3.2 Storage requirement 225
12.9.1.3.3 Storage management 225
12.9.1.4 CRL file location and naming convention 225
12.9.1.5 Operational model 226
12.9.1.6 Examples 226
12.9.1.6.1 Revocation of a broadcaster's certificate 226
12.9.1.6.2 Revocation of a CA's certificate. 226
12.9.1.7 CRL format 226
12.9.1.8 Profile of CRL 227
12.9.1.9 CRL Processing 227
12.9.2 Root certificate management 228
12.9.2.1 Introduction 228
12.9.2.2 Security of RCMM 228
12.9.2.3 Format of RCMM 229
12.9.2.4 Distribution of RCMM 229
12.9.2.5 RCMM Processing 229
12.9.2.6 Example: Renewal of a root certificate 230
12.9.3 Test certificates 231
12.10 Security on the return channel 231
12.10.1 GEM application functionality 231
12.10.2 TLS cipher suites 231
12.10.3 Downloading of certificates for TLS 232
12.10.3.1 Introduction 232
12.10.3.2 Usage of certificate in TLS 232
12.10.3.2.1 When certificates are delivered with the application 232
12.10.3.2.2 When no certificates are provided 233
12.10.3.2.3 CRL distribution points 233
12.11 The internet profile of X.509 (informative) 233
12.11.1 Main part of the certificate 233
12.11.1.1 Certificate 233
12.11.1.2 signatureAlgorithm 233
12.11.1.3 signatureValue 234
12.11.1.4 tbsCertificate 234
12.11.1.5 version 234
12.11.1.6 serialNumber 235
12.11.1.7 signature 235
12.11.1.8 issuer 235
12.11.1.9 validity 235
12.11.1.9.1 UTCTime 236
12.11.1.9.2 GeneralizedTime 236
12.11.1.10 subject 236
12.11.1.10.1 issuerUniqueID 236
12.11.1.10.2 subjectUniqueID 236
12.11.1.11 SubjectPublic Key Info 237

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 14 of 861

12.11.1.12 Unique Identifiers 237
12.11.1.13 Extensions 237
12.11.2 Standard certificate extensions 238
12.11.2.1 Authority key identifier 238
12.11.2.2 Subject key identifier 238
12.11.2.3 Key usage 238
12.11.2.4 Private key usage period 238
12.11.2.5 Certificate policies 238
12.11.2.6 Policy mappings 239
12.11.2.7 Subject Alternative Name 239
12.11.2.8 Issuer Alternative Name 239
12.11.2.9 Subject Directory attributes 239
12.11.2.10 Basic Constraints 240
12.11.2.11 Name Constraints 240
12.11.2.12 Policy Constraints 240
12.11.2.13 Extended key usage field 240
12.11.2.14 CRL Distribution points 241
12.12 Platform minima 241
12.13 Plug-ins 241
12.14 Applications loaded from an interaction channel 242
12.14.1 Permission for application loading from the return channel 242
12.15 Stored applications 242
12.15.1 Stand-alone stored applications 243
12.15.2 Cached applications 243
12.16 Void 243
12.17 Authentication of unbound applications 243
12.18 Authentication of privileged applications 243

13 Graphics reference model 244
13.1 Introduction 244
13.1.1 Interapplication interaction 244
13.2 General Issues 245
13.2.1 Coordinate Spaces 245
13.2.1.1 Normalized screen space 245
13.2.1.2 User space 246
13.2.1.3 Pixel Aspect Ratio 248
13.2.1.4 Video space 249
13.2.2 Dependencies between HAVi screen device configurations (informative) 249
13.2.2.1 Principles 249
13.2.2.2 Usage examples for standard definition, 625-line markets 250
13.2.2.3 High definition usage examples 250
13.2.2.4 Scalability and extensibility 251
13.3 Graphics 251
13.3.1 Modelling of the GEM display stack composition 251
13.3.2 AWT Reference Model in the GEM 253
13.3.3 HAVi devices and AWT components 254
13.3.3.1 Video and graphics pixel aligned 255
13.3.3.2 Zero graphics impact 256
13.3.4 Composition 256
13.3.4.1 AWT paint rule 256
13.3.5 Composition Rules 257
13.3.5.1 Components generally 257
13.3.6 Extensions to the AWT graphics capabilities 257
13.3.6.1 Graphics Objects in GEM applications 258
13.3.6.2 Buffered Image 258
13.3.6.3 DVBColor 258
13.3.6.3.1 Modified packed colour representation 258
13.3.7 14:9 Aspect Ratio Support 259
13.3.8 Interaction between graphics from GEM and resident applications 259
13.4 Video 259
13.4.1 Component-based players and background players 259
13.4.2 Modelling MPEG decoding and presentation pipeline 260

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 15 of 861

13.4.3 Coordinate Spaces 261
13.4.4 Video components 261
13.5 Subtitles 262
13.5.1 Language and presentation setting 262
13.5.2 Relation to graphics 263
13.5.3 Coordinate Spaces 263
13.6 Approximations 263
13.6.1 Approximations in composition 263
13.6.1.1 Implementation of modes 264
13.6.1.1.1 Graphics directly over video 264
13.6.1.1.2 Graphics over other graphics 264
13.6.1.2 Approximation of alpha 265
13.6.1.3 Approximation of colour 266

14 System integration aspects 267
14.1 Namespace mapping 267
14.2 Reserved names 267
14.3 XML notation 267
14.4 Network signalling (error behaviour) 269
14.5 Text encoding of application identifiers 269
14.6 Filename requirements 270
14.6.1 Persistent storage 270
14.6.2 DSMCC object carousel 270
14.6.3 Stored applications 271
14.7 Files and file names 271
14.8 Locators and content referencing 272
14.9 Content referencing for IPTV/ OTT 272
14.9.1 Introduction (informative) 272
14.9.2 Details 273
14.10 Service identification 273
14.11 CA system 274
14.12 Focus management 274

15 Detailed platform profile definitions 275
15.1 PNG - restrictions 278
15.2 Minimum media formats supported by DVB-J APIs 278
15.3 JPEG - restrictions 279
15.4 Locale support 279
15.5 Video raster format dependencies 279
15.5.1 Standard Definition (PAL/SECAM or NTSC resolution) 279
15.5.1.1 Logical pixel resolution 279
15.6 Functional equivalents 279
15.6.1 Modifications to MHP Definitions of Functional equivalents 281
15.6.1.1 Carousel 281
15.6.1.1.1 NSAP Address 281
15.6.1.1.2 Content type descriptor 281
15.6.1.1.3 Application Icons Descriptor 281
15.6.1.2 Application Signalling 282
15.6.1.2.1 Transport protocol descriptor 282
15.6.1.2.2 AIT descriptor tag values 282
15.6.1.3 Application Name Descriptor 282

16 Registry of constants 283
16.1 System constants 283
16.2 DVB-J constants 283
16.2.1 Public and Protected final static primitive fields from DVB packages 283

17 Internet access clients 286
17.1 Referencing DVB services and content within WWW content 286
17.2 Minimum requirements for internet clients 286
17.3 Internet streamed media 286
17.4 Internet connection management 286

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Annex A (normative): External references; errata, clarifications and exemptions 288
A.1 Void....288
A.2 Errata to DAVIC 288
A.2.1 org.davic.media.MediaTimeEventControl - deregistering listeners....288
A.2.2 org.davic.resources.ResourceClient.requestRelease 289
A.2.3 org.davic.mpeg....289
A.2.3.1 General....289
A.2.3.2 NotAuthorizedException....289
A.2.4 Chapter 9, "Application Format"....291
A.2.4.1 Section 9.4.7, "The MPEG-2 Section Filter API" 291
A.2.5 org.davic.mpeg.sections 291
A.2.5.1 RingSectionFilter 291
A.2.5.2 Section....291
A.2.5.2.1 clone() 291
A.2.5.2.2 getData()....291
A.2.5.2.3 getFullStatus()....291
A.2.5.2.4 Class Description 292
A.2.5.3 SectionFilter 292
A.2.5.3.1 Cross reference error 292
A.2.5.3.2 startFiltering(all signatures) 292
A.2.5.3.3 startFiltering(java.lang.Object, int, int, int, byte[], byte[])....292
A.2.5.3.4 startFiltering (appData, pid, tableId) exceptions 292
A.2.5.3.5 Started Section Filters 292
A.2.5.4 SectionFilterGroup....292
A.2.5.4.1 attach....292
A.2.5.4.2 Constructors 292
A.2.5.4.3 sectionSize 293
A.2.5.4.4 newRingSectionFilter....293
A.2.5.4.5 Constructor(int, boolean) 293
A.2.5.4.6 General....293
A.2.5.5 TimeOutEvent....293
A.2.6 Simple Section Filter....294
A.2.7 org.davic.media....294
A.2.7.1 FreezeControl.resume()....294
A.2.7.2 MediaTimePositionChangedEvent....294
A.2.7.3 NotAuthorizedMediaException 295
A.2.7.4 LanguageControl....296
A.2.7.4.1 Class description 296
A.2.7.4.2 selectLanguage(String) 296
A.2.8 org.davic.net....296
A.2.8.1 InvalidLocatorException....296
A.2.8.2 Locator....297
A.2.8.2.1 Locator()....297
A.2.8.2.2 toExternalForm()....297
A.2.8.3 tuning 297
A.2.8.3.1 NetworkInterfaceController 297
A.2.8.3.1.1 reserve() 297
A.2.8.3.1.2 reserveFor()....298
A.2.8.3.1.3 tune()....298
A.2.8.3.2 NetworkInterface....298
A.2.8.3.2.1 Protected constructor....298
A.2.9 org.davic.net.tuning....298
A.2.9.1 Figure H-1....298
A.2.9.2 Figure H-2....298
A.2.9.3 Network Interface....298
A.2.10 Extensibility and Over-Riding 299
A.3 Additional GEM requirements on Java TV....299
A.3.1 javax.tv.util.TVTimerSpec....299
A.3.2 javax.media.Manager 299
A.3.2.1 getDataSourceList() 299

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 17 of 861

A.3.2.2 getHandlerClassList() 299
A.4 HAVi 300
A.4.1 Drafting conventions 300
A.4.2 General 300
A.4.2.1 Thread-safety 300
A.4.2.2 javadoc errors 300
A.4.3 Event mechanism 301
A.4.3.1 Introduction 301
A.4.3.2 Overview of HAVi events 301
A.4.3.3 Relation between HAVi events and AWT events 303
A.4.3.4 Application guidelines 303
A.4.4 org.havi.ui 303
A.4.4.1 HActionable 303
A.4.4.1.1 getActionSound 303
A.4.4.1.2 Class description 303
A.4.4.1.3 setActionCommand 303
A.4.4.2 HActionInputPreferred 304
A.4.4.3 HAdjustmentInputPreferred 304
A.4.4.3.1 Interface description 304
A.4.4.3.2 getAdjustMode 304
A.4.4.3.3 proccessHAdjustmentEvent 304
A.4.4.3.4 setAdjustMode 304
A.4.4.4 HAdjustmentValue 305
A.4.4.4.1 Class description 305
A.4.4.4.2 getAdjustmentSound 305
A.4.4.4.3 getBlockIncrement 305
A.4.4.4.4 getUnitIncrement 305
A.4.4.5 HAnimateEffect 305
A.4.4.5.1 getRepeatCount 305
A.4.4.6 HBackgroundDevice 305
A.4.4.6.1 setBackgroundConfiguration 305
A.4.4.6.2 getConfigurations 305
A.4.4.6.3 Constructor 306
A.4.4.7 HBackgroundImage 306
A.4.4.7.1 Constructors - HBackgroundImage(String), HBackgroundImage(URL) 306
A.4.4.7.2 load 306
A.4.4.7.3 Constructor(byte[]) 306
A.4.4.8 HComponent 306
A.4.4.8.1 processEvent 306
A.4.4.9 HComponentOrdering 306
A.4.4.9.1 addAfter, addBefore 306
A.4.4.10 HEventMulticaster 307
A.4.4.10.1 Class description 307
A.4.4.10.2 Constructor 307
A.4.4.10.3 remove 307
A.4.4.11 HFontCapabilities 307
A.4.4.11.1 getSupportedCharacterRanges 307
A.4.4.12 HGraphicsDevice 308
A.4.4.12.1 getBestConfiguration 308
A.4.4.12.2 setGraphicsConfiguration 308
A.4.4.12.3 getConfigurations 308
A.4.4.12.4 Constructor 308
A.4.4.13 HGraphicsConfigTemplate 308
A.4.4.13.1 setPreference 308
A.4.4.14 HListElement 309
A.4.4.14.1 Class description 309
A.4.4.15 HListGroup 309
A.4.4.15.1 Class description 309
A.4.4.15.2 Fields 309
A.4.4.15.2.1 ITEM_NOT_FOUND 309
A.4.4.15.2.2 ADD_INDEX_END 310

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 18 of 861

A.4.4.15.3 Use of "action" and "actioning" 310
A.4.4.15.4 addItem(HListItem item, int index) 310
A.4.4.15.5 addItem(HListItem items[], int index) 310
A.4.4.15.6 getIconSize 310
A.4.4.15.7 getLabelSize 310
A.4.4.15.8 getOrientation 310
A.4.4.15.9 setFocusTraversal 311
A.4.4.15.10 setItemSelected 311
A.4.4.15.11 setIconSize 311
A.4.4.15.12 setLabelSize 311
A.4.4.15.13 setListContent 311
A.4.4.15.14 setMove 312
A.4.4.15.15 setScrollPosition 312
A.4.4.15.16 setSelectionMode 312
A.4.4.15.17 removeItem 312
A.4.4.15.18 removeAlltems 312
A.4.4.15.19 setCurrentItem 313
A.4.4.15.20 setMultiSelection 313
A.4.4.16 HListGroupLook 313
A.4.4.16.1 getMaximumSize 313
A.4.4.16.2 getMinimumSize 314
A.4.4.16.3 getPreferredSize 314
A.4.4.16.4 getValue 315
A.4.4.16.5 hitTest 315
A.4.4.16.6 showLook 315
A.4.4.16.7 Class description 316
A.4.4.16.8 showLook and renderVisible 316
A.4.4.16.9 renderVisible 317
A.4.4.17 HLook 317
A.4.4.17.1 General 317
A.4.4.17.2 Class description 317
A.4.4.17.3 getPreferredSize 317
A.4.4.17.4 showLook 317
A.4.4.18 HMultilineEntry 317
A.4.4.19 HMultilineEntryLook 318
A.4.4.19.1 getCaretCharPositionForLine 318
A.4.4.19.2 getSoftLineBreakPositions 318
A.4.4.19.3 getVisibleSoftLineBreakPositions 318
A.4.4.20 HNavigable 318
A.4.4.20.1 Class description 318
A.4.4.20.2 setMove 318
A.4.4.20.3 set FocusTraversal 318
A.4.4.21 HOrientable 318
A.4.4.21.1 Interface description 318
A.4.4.21.2 getOrientation 319
A.4.4.21.3 setOrientation 319
A.4.4.21.3.1 Orientation constants 319
A.4.4.22 HScene 319
A.4.4.22.1 addAfter, addBefore 319
A.4.4.22.2 getFocusOwner 319
A.4.4.22.3 Rendering behavior 319
A.4.4.22.4 show 319
A.4.4.22.5 setVisible 320
A.4.4.23 HScreenConfigurationListener 320
A.4.4.24 HSceneFactory 320
A.4.4.24.1 getDefaultHScene 320
A.4.4.24.2 Class Description 320
A.4.4.25 HSelectionInputPreferred 320
A.4.4.25.1 Interface description 320
A.4.4.25.2 getSelectionMode 321
A.4.4.25.3 processHItemEvent 321
A.4.4.25.4 setSelectionMode 321

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.26 HSingleLineEntry 321
A.4.4.26.1 Class description 321
A.4.4.26.2 Default parameter values not exposed in the constructors 321
A.4.4.26.3 setType(int) 321
A.4.4.26.4 getValidInput 322
A.4.4.27 HStaticAnimation 322
A.4.4.28 HStaticRange 322
A.4.4.28.1 Fields 322
A.4.4.28.1.1 SCROLLBAR_BEHAVIOR 322
A.4.4.28.1.2 SLIDER_BEHAVIOR 322
A.4.4.28.2 getOrientation 322
A.4.4.28.3 setBehavior 323
A.4.4.28.4 setRange 323
A.4.4.28.5 setThumbOffsets 323
A.4.4.28.6 setValue 323
A.4.4.29 HVideoDevice 323
A.4.4.29.1 getBestConfiguration 323
A.4.4.29.2 setVideoConfiguration 323
A.4.4.29.3 getVideoController 324
A.4.4.29.4 setVideoConfiguration 324
A.4.4.29.5 getConfigurations 324
A.4.4.29.6 getVideoSource 324
A.4.4.29.7 Constructor 324
A.4.4.30 HVisible 324
A.4.4.30.1 Class description 324
A.4.4.30.2 Constructors 325
A.4.4.30.3 setDefaultSize 325
A.4.4.30.4 setLookData 325
A.4.4.30.5 setResizeMode 325
A.4.4.30.6 setTextContent 325
A.4.4.30.7 NO_DEFAULT_SIZE 326
A.4.4.31 HVideoConfigTemplate 326
A.4.4.31.1 setPreference 326
A.4.4.32 HVideoComponent 326
A.4.4.32.1 HAVi Specification 326
A.4.4.32.2 removeOnScreenLocationModifiedListener 326
A.4.4.33 HBackgroundConfiguration 326
A.4.4.33.1 setColor 326
A.4.4.33.2 getConfigTemplate 326
A.4.4.33.3 Constructor 327
A.4.4.34 HScreenDevice 327
A.4.4.34.1 reserveDevice 327
A.4.4.35 HImageMatte 327
A.4.4.35.1 setMatteData 327
A.4.4.36 HVersion 327
A.4.4.36.1 General 327
A.4.4.36.2 MHP Specific Clarification 327
A.4.4.36.3 Field descriptions 328
A.4.4.37 HKeyboardInputPreferred 328
A.4.4.37.1 INPUT_NUMERIC 328
A.4.4.37.2 INPUT_ALPHA 328
A.4.4.37.3 getValidInput 328
A.4.4.38 HScreenConfigTemplate 329
A.4.4.38.1 Class description 329
A.4.4.38.2 VIDEO_GRAPHICS_PIXEL_ALIGNED 329
A.4.4.38.3 DISABLED 329
A.4.4.38.4 SCREEN_ASPECT_RATIO 329
A.4.4.39 HGraphicsConfiguration 330
A.4.4.39.1 getConfigTemplate 330
A.4.4.39.2 Constructor 330
A.4.4.40 HVideoConfiguration 330
A.4.4.40.1 getConfigTemplate 330

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.40.2 Constructor 330
A.4.4.41 HToggleButton 330
A.4.4.41.1 Default parameter values exposed in the constructors 330
A.4.4.42 HGraphicButton 331
A.4.4.42.1 Default parameter values exposed in the constructors 331
A.4.4.43 HTextButton 331
A.4.4.43.1 Default parameter values exposed in the constructors 331
A.4.4.44 HLook and classes implementing HLook 331
A.4.4.45 HTextLook 331
A.4.4.45.1 General 331
A.4.4.46 HExtendedLook 332
A.4.4.47 HAnimateLook, HGraphicLook, HListGroupLook, HMultilineEntryLook, HRangeLook, HSinglelineEntryLook and HTextLook 334
A.4.4.48 HSwitchable 334
A.4.4.49 HStillImageBackgroundConfiguration 334
A.4.4.49.1 Constructor 334
A.4.4.50 HTextValue 335
A.4.4.50.1 Class description 335
A.4.4.51 HDefaultTextLayoutManager 335
A.4.4.51.1 getMinimumSize 335
A.4.4.51.2 getMaximumSize 335
A.4.4.51.3 getPreferredSize 335
A.4.4.52 Hscreen 335
A.4.4.52.1 getHGraphicsDevices, getHVideoDevices, getHBackgroundDevices 335
A.4.4.52.2 getCoherentScreenConfigurations 336
A.4.4.52.3 setCoherentScreenConfigurations 336
A.4.4.52.4 Additional methods 336
A.4.5 org.havi.ui.event 337
A.4.5.1 HActionEvent 337
A.4.5.1.1 getModifiers 337
A.4.5.2 HItemEvent 337
A.4.5.2.1 Constructor 337
A.4.5.2.2 ITEM_SET_CURRENT 337
A.4.5.3 HKeyEvent 337
A.4.5.3.1 Constructors 338
A.4.5.4 HRcEvent 338
A.4.5.5 HRcCapabilities 338
A.4.5.5.1 getRepresentation 338
A.4.5.5.2 Constructor 338
A.4.5.6 HEventGroup 338
A.4.5.6.1 getKeyEvents 338
A.4.5.7 HKeyCapabilities 339
A.4.5.7.1 Constructor 339
A.4.5.8 HEventRepresentation 339
A.4.5.8.1 Constructor 339
A.4.6 References to ISO/IEC10646-1:1993 339
A.4.7 Chapter 8 339
A.4.7.1 Section 8.2.3.3.2 "Compatibility with Existing java.awt Methods" 339

A.5 ISO/IEC 13818-6 339
A.5.1 Reconstruction of NPT 339

Annex B (normative): Broadcast filesystem and trigger transport 340

B.0 General 340

B.1 Service domain 340

B.2 Filesystem requirements 340
B.2.1 Static requirements 340
B.2.1.1 Caching behaviour 341
B.2.2 Filesystem updates 341
B.2.3 Content type descriptor 341

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

B.3 Stream description 342
B.4 Trigger signalling 342
B.4.0 General 342
B.4.1 Trigger object 342
B.4.2 Trigger event 343
B.4.2.1 Extrapolation of timebase values 343
B.4.2.2 Monitoring of trigger events 344
B.5 Caching 344
B.5.1 Determining file version 344
B.5.2 Transparency levels of caching 344
B.5.2.1 Transparent caching 344
B.5.2.1.1 Active caching 345
B.5.2.1.2 Passive caching 345
B.5.2.1.3 DII repetition rate 345
B.5.2.2 Semi-transparent caching 345
B.5.2.2.1 Implications for the terminal (informative) 346
B.5.2.3 Static caching 346
B.5.2.3.1 Implications for the broadcaster (informative) 346
B.5.2.3.2 Implications for the terminal (informative) 346
B.5.3 Dynamic carousel structure 346
Annex C (informative): Bibliography 347
C.1 Television 347
C.2 Java 347
C.3 Internet and HTTP 348
C.4 Other 348
Annex D (normative): Text presentation 350
D.1 Scope 350
D.2 Fonts 350
D.2.1 Embedded fonts 350
D.2.2 Downloaded fonts 350
D.2.2.1 Font technology 350
D.2.2.2 Font index files 350
D.2.2.2.1 Format of file 350
D.2.2.2.2 Element semantics 351
D.2.2.2.3 Example 352
D.2.2.3 Name and location of font index files 352
D.2.2.3.1 General 352
D.2.2.3.2 Name of file 352
D.2.2.3.3 Location 352
D.2.2.4 Specification of fonts at run time 352
D.2.2.4.1 DVB-J 352
D.3 Text rendering 353
D.3.1 Low and high level rendering 353
D.3.1.1 Low level rendering 353
D.3.1.2 High level rendering 353
D.3.2 Philosophy 353
D.3.2.1 High level rendering conceptual process 353
D.3.3 Font Definition 354
D.3.3.1 Font bounds 354
D.3.3.2 "Physical" font data 355
D.3.3.3 Ligatures and Glyph Substitution 355
D.3.4 Converting font metrics to display pixels 355
D.3.4.1 Vertical resolution 355
D.3.4.2 Horizontal resolution 356

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

D.3.5 Rendering within limits and insets 356
D.3.5.1 Low level rendering 356
D.3.5.2 High level rendering 357
D.3.5.3 Conversion of units 357
D.3.5.3.1 yOffsetTop 357
D.3.5.3.2 yOffsetBottom 357
D.3.5.3.3 xOffsetLeft 357
D.3.5.3.3.1 With 14:9 graphics 357
D.3.5.3.3.2 With 4:3 graphics 357
D.3.5.3.3.3 With 16:9 graphics 358
D.3.5.3.4 outlineResolution 358
D.3.5.3.5 Font bounds 358
D.3.6 "logical" text width rules 358
D.3.6.1 Computing "logical" text width 359
D.3.6.1.1 Font sizes 359
D.3.6.1.2 Character widths 359
D.3.6.1.3 Kerning 359
D.3.6.1.4 Letter spacing 360
D.3.6.2 Logical text width 360
D.3.7 Line breaking 360
D.3.7.1 Text wrapping setting is false 360
D.3.7.2 Text wrapping setting is true 360
D.3.8 Positioning lines of text vertically within an object 362
D.3.8.1 Number of lines 362
D.3.8.2 Truncation 362
D.3.8.3 Positioning 362
D.3.8.3.1 Vertical alignment setting is VERTICAL_START_ALIGN 362
D.3.8.3.2 Vertical alignment setting is VERTICAL_END_ALIGN 362
D.3.8.3.3 Vertical alignment setting is VERTICAL_CENTER 362
D.3.8.3.4 Examples 363
D.3.9 Rendering lines of text horizontally 364
D.3.9.1 Available width 364
D.3.9.2 Truncation 364
D.3.9.3 Placement 364
D.3.10 Text overflow 365
D.3.11 Tabulation 365
D.3.12 Placing runs of characters and words 366
D.3.13 Control of text flow 366

D.4 Text mark-up 366
D.4.1 White Space Characters 366
D.4.2 Marker characters 367
D.4.3 Non-printing characters 367
D.4.4 Format Control Mark-up 367
D.4.5 Future compatibility 368

Annex E (normative): Character set 369

Annex F (informative): Authoring and implementation guidelines 375

F.1 Authoring Guidelines 375

F.2 Implementation Guidelines 375

F.3 Authoring guidelines for DVB-J 375

Annex G (normative): Minimum platform capabilities 376

G.1 Graphics 376
G.1.1 Device capabilities 376
G.1.1.1 General 376
G.1.1.2 Standard definition 376
G.1.1.3 High definition 377
G.1.2 Video presentation capabilities 379

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

G.1.3 Image processing capabilities 380
G.1.3.1 Composition rules 380
G.1.4 Alpha capabilities 380
G.1.5 Colour capabilities 380
G.1.6 MPEG I frame and Video drips 380
G.2 Audio 381
G.3 Video 381
G.4 Resident fonts and text rendering 381
G.4.1 The built-in font 381
G.4.2 Presentation to DVB-J 382
G.4.3 Text directions 382
G.5 Input events 382
G.6 Memory 384
G.7 Other resources 385
Annex H (normative): Extensions 387
Annex I (normative): DVB-J fundamental classes 388
Annex J (normative): DVB-J event API 391
J.1 Overview 391
J.2 The resource management 393
J.3 The Event Repository 393
J.3.1 Example 393
J.4 Unicode 394
J.5 Virtual keyboards 394
Annex K (normative): DVB-J persistent storage API 409
Annex L (normative): User settings and preferences API 414
Annex M (informative): SI access API 425
Annex N (normative): Streamed media API extensions 426
N.1 Active Format Definition 426
N.1.1 MHP Signalling for Active Format Definition 426
N.1.2 Drip-feed APIs 426
N.1.3 VideoFormatControl 426
N.2 Streaming monitoring API 463
N.3 Media Stream Synchronization API 470
Annex O (normative): Integration of the JavaTV SI API 475
Annex P (normative): Broadcast transport protocol access 476
P.1 Overview 476
P.2 The org.dvb.dsmcc package 477
P.2.0 General 477
P.2.1 DSMCCObject 477
P.2.1.1 DSMCCObject.getSigners() 477
P.2.1.2 DSMCCObject.getSigners(boolean known_root) 477
P.2.2 DSMCCStream 477
P.2.2.1 isAudio() method 477
P.2.2.2 isData() method 477

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

P.2.2.3 isMPEGProgram() method 478
P.2.2.4 isVideo() method 478
P.2.3 DSMCCStreamEvent 478
P.2.3.1 Lightweight binding of trigger API 478
P.2.3.1.1 DSMCCStreamEvent.getEventList() 478
P.2.3.1.2 StreamEvent.getEventId() 478
P.2.3.1.3 DSMCCStreamEvent.unsubscribe(int, StreamEventListener) 478
P.2.4 InvalidFormatException 478
P.2.5 ServiceDomain 478
P.2.5.1 ServiceDomain.attach(byte[]) 479
P.2.5.2 ServiceDomain.attach 479
P.2.5.2.1 ServiceDomain.attach(Locator) 479
P.2.5.2.2 ServiceDomain.attach(Locator, int) 479
P.2.5.3 ServiceDomain.getLocator() 479
P.2.5.4 ServiceDomain.getNSAPAddress() 479
P.2.5.5 ServiceDomain.getURL(Locator) 479
P.2.5.6 ServiceDomain.isNetworkConnectionAvailable() 480
P.2.6 ServiceXFRErrorEvent 480
P.2.7 ServiceXFRException 480
P.2.8 ServiceXFRReference 480
P.2.9 StreamEvent 480
P.3 Support for Stored Applications 480
P.4 Javadoc for org.dvb.dsmcc package 481
Annex Q (normative): Datagram socket buffer control 532
Annex R (normative): DVB-J return channel connection management API 534
Annex S (normative): Application listing and launching 554
Annex T (normative): Permissions 584
Annex U (normative): Extended graphics APIs 589
Annex V: Void632
Annex W (informative): DVB-J examples 633
W.1 DVB-J examples from MHP 633
W.2 Example of enumeration extension 633
W.3 Example of testing for optional APIs 634
W.4 Example of lightweight trigger API 636
W.5 Example of media stream synchronization API 636
Annex X (normative): Test support 639
Annex Y (normative): Inter-application and Inter-Xlet communication API 647
Annex Z (informative): Services, service contexts and applications in a GEM environment 652
Z.1 Introduction 652
Z.2 Basic concepts 652
Z.3 Presenting a service in GEM 652
Z.3.1 Presenting the media components of a service 652
Z.3.2 Presenting the application components of a service 653
Z.4 Service Context Object Design 654
Z.4.1 Overview 654
Z.4.2 Single-Application Model 654

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Z.4.3 Multiple-Application Model 655
Z.4.4 Considerations for standalone stored services 655
Z.5 Multiple service contexts in a GEM platform 656
Z.6 How does the platform know which services are available? 656
Annex AA: Void 657
Annex AB: Void 658
Annex AC: Void 659
Annex AD: Void 660
Annex AE (normative): Inner Applications 661
Annex AF (normative): Plug-in APIs 672
Annex AG (normative): Stored application APIs 680
Annex AH (normative): Internet client APIs 706
Annex AI (normative): DVB Extensions for cryptography 728
Annex AJ (normative): Cryptographics service provider installation 746
AJ.1 Introduction (informative) 746
AJ.2 The org.dvb.security.provider package 746
Annex AK (normative): Extended service selection API 749
Annex AL (normative): Extended content referencing API 751
Annex AM (normative): Smart card reader API 754
Annex AN (normative): Provider APIs 759
Annex AO (normative): Services and the service list 800
AO.1 Service list creation 800
AO.2 SIManager 800
AO.2.1 Introduction (informative) 800
AO.2.2 Method definition 801
AO.2.2.1 Methods without modified or extended semantics 801
AO.2.2.2 getTransports 801
AO.2.2.3 getSupportedDimensions 801
AO.3 Services 801
AO.3.1 Services only signalled in TS Full SI Broadcast Discovery Information Record 802
AO.3.1.1 getLocator 802
AO.3.1.2 getName 802
AO.3.1.3 getServiceType 802
AO.3.1.4 retrieveDetails 802
AO.3.1.5 hasMultipleInstances 802
AO.3.2 Services only signalled in TS Partial SI Broadcast Discovery Information Record 802
AO.3.2.1 General 802
AO.3.2.2 getLocator 803
AO.3.2.3 getName 803
AO.3.2.4 getServiceType 803
AO.3.2.5 retrieveDetails 803
AO.3.2.6 hasMultipleInstances 803
AO.3.3 Services signalled in more than one mechanism 803
AO.3.3.1 getLocator 803

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

AO.3.3.2 getName 803
AO.3.3.3 getServiceType 804
AO.3.3.4 retrieveDetails 804
AO.3.3.5 hasMultipleInstances 804

Annex AP (normative): Mapping between Java TV and service discovery and selection 805

AP.1 Introduction (informative) 805

AP.2 Generic issues 805
AP.2.1 dvb:MultilingualType 805
AP.2.2 Information available only when a stream is actually received 805

AP.3 Mappings independent of Discovery Information Record Type 806
AP.3.1 javax.tv.service.navigation.ServiceProviderInformation 806
AP.3.2 javax.tv.service.navigation.StreamType 806
AP.3.3 javax.tv.service.navigation.ServiceComponent 806
AP.3.4 javax.tv.service.transport.Network 806
AP.3.5 javax.tv.service.transport.Transport 806
AP.3.5.1 getDeliverySystemType 806
AP.3.6 javax.tv.service.transport.Bouquet 806
AP.3.6.1 getBouquetID 806
AP.3.6.2 getName 806
AP.3.7 javax.tv.service.transport.BouquetCollection 807
AP.3.8 javax.tv.service.transport.TransportStream 807
AP.3.9 javax.tv.service.transport.TransportStreamCollection 807
AP.3.10 javax.tv.service.transport.NetworkCollection 807
AP.3.11 javax.tv.service.ServiceInformationType 807
AP.3.12 javax.tv.service.ServiceType 807
AP.3.13 javax.tv.service.navigation.DeliverySystemType 807
AP.3.14 ServiceNumber 807
AP.3.15 javax.tv.service.navigation.CAIdentification 807
AP.3.16 javax.tv.service.navigation.FavoriteServicesName 807
AP.3.17 javax.tv.service.transport.Network 808

AP.4 Mappings specific to TS Full SI Broadcast Discovery Information Record 808
AP.4.1 javax.tv.service.navigation.ServiceDetails 808

AP.5 Mappings specific to TS Partial SI Broadcast Discovery Information Record 808
AP.5.1 javax.tv.service.navigation.ServiceDetails 808
AP.5.1.1 getDeliverySystemType 808
AP.5.1.2 getLongName 808
AP.5.1.3 getService 808
AP.5.1.4 getLocator 808
AP.5.1.5 getServiceType 809
AP.5.1.6 retrieveComponents 809
AP.5.1.7 retrieveServiceDescription 809
AP.5.1.8 addServiceComponentChangeListener 809
AP.5.2 javax.tv.service.navigation.ServiceDescription 809
AP.5.3 javax.tv.service.navigation.ServiceComponent 809
AP.5.3.1 getName 809
AP.5.3.2 getAssociatedLanguage 809
AP.5.3.3 getStreamType 809
AP.5.3.4 getService 809

Annex AQ (normative): Mapping between Java TV and broadband content guide 810

AQ.1 Finding the BCG Provider 810

AQ.2 javax.tv.service.Service and javax.tv.service.navigation.ServiceDetails 810

AQ.3 javax.tv.service.guide.ProgramSchedule 810
AQ.3.1 retrieveCurrentProgramEvent 810
AQ.3.2 retrieveFutureProgramEvents 810
AQ.3.3 retrieveNextProgramEvent 810

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 27 of 861

AQ.3.4 retrieveProgramEvent 810
AQ.3.5 addListener 811
AQ.3.6 removeListener 811
AQ.3.7 getServiceLocator 811

AQ.4 javax.tv.service.guide.ProgramEvent 811
AQ.4.1 getLocator 811
AQ.4.2 getStartTime 811
AQ.4.3 getEndTime 811
AQ.4.4 getDuration 811
AQ.4.5 getName 812
AQ.4.6 retrieveDescription 812
AQ.4.7 getRating 812
AQ.4.8 getService 812
AQ.4.9 retrieveComponents 812

AQ.5 javax.tv.service.guide.ProgramEventDescription 812

Annex AR (normative): XML encoding for AIT 813

AR.1 Introduction 813

AR.2 Extensions to defined SD&S elements 813
AR.2.1 Package 813
AR.2.2 IP Service 814

AR.3 New XML element definitions 814
AR.3.1 ApplicationList 814
AR.3.2 Application 814
AR.3.3 ApplicationIdentifier 815
AR.3.4 ExternalApplicationIdentifier 815
AR.3.5 ApplicationDescriptor 816
AR.3.6 VisibilityDescriptor 817
AR.3.7 IconDescriptor 817
AR.3.8 AspectRatio 817
AR.3.9 MhpVersion 818
AR.3.10 StorageCapabilities 818
AR.3.11 StorageType 819
AR.3.12 ApplicationType 819
AR.3.13 DvbApplicationType 819
AR.3.14 ApplicationControlCode 820
AR.3.15 ApplicationSpecificDescriptor 820
AR.3.16 DVBJDescriptor 820
AR.3.17 ApplicationStructure 821
AR.3.18 AbstractIPService 821
AR.3.19 UnboundApplicationDescriptor 822
AR.3.20 Provider descriptors 822

Annex AS (informative): IPTV Use-cases 824

AS.1 Simple use-case 824

Annex AT (normative): Application Management API 825

Annex AU (normative): AU.1. IPTV content referencing API 829

AU.2. OTT content referencing API 833

Annex AV (normative): Extended service list API 835

Annex AW (normative): API to DVB service discovery and selection 843

Annex AX (normative): API to DVB broadband content guide 846

Annex AY (normative): TV-Anytime and Java TV Integration 851

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Annex AZ (normative): MHP terminal hardware API 854
Annex BA (normative): Content Download API 856
History 860

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Intellectual Property Rights

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

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

Foreword

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

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

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

The Digital Video Broadcasting Project (DVB) is an industry-led consortium of broadcasters, manufacturers, network operators, software developers, regulatory bodies, content owners and others committed to designing global standards for the delivery of digital television and data services. DVB fosters market driven solutions that meet the needs and economic circumstances of broadcast industry stakeholders and consumers. DVB standards cover all aspects of digital television from transmission through interfacing, conditional access and interactivity for digital video, audio and data. The consortium came together in 1993 to provide global standardisation, interoperability and future proof specifications.

Introduction

Purpose

The DVB system already provides a comprehensive toolbox to enable interoperable digital video broadcasting systems based on MPEG-2 standards for various transmission media including satellite, cable, terrestrial and microwave. This toolbox also covers interactive services using different kinds of return channels and further supporting functionalities such as service information and many others.

The Globally Executable Multimedia Home Platform (GEM) adds a technical solution for the user terminal that enables the reception and presentation of applications in a vendor, author and broadcaster neutral framework. Here "neutral" includes scenarios that consider legacy infrastructure. Applications from various service providers will be interoperable with different GEM implementations in a horizontal market, where applications, networks, and GEM terminals can be made available by independent providers.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Aim

The aim of the present document is to encourage implementations of:

- receivers and middleware,
- applications,
- conformance tests.

DVB welcomes feedback from implementers of these. DVB reserves the right to publish a subsequent revision of the present document with (possibly significant) changes based on that feedback.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

1 Scope

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

The present document is firstly intended to be used by entities writing terminal specifications and/or standards based on GEM. 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 [1], Blu-ray [13], OCAP [5], ACAP [7], ARIB [6], and the Open IPTV Procedural Application Gateway [8]. Implementers should consult the publisher of specifications which reference GEM regarding conformance.

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

Since the current business models for delivery of DVB services via broadband IP networks are more operator controlled than was assumed for the original GEM, the present document extends GEM with support for operator supplied applications which run all the time that the GEM environment is running. There is also the possibility that where an operator has subsidised the GEM terminal, applications from that operator may have specially privileges not available to normal applications.

Clauses 1 to 14 specify the applicable technologies and technical definitions in a generic way. Clause 15 provides detailed profile definitions for the following initial profiles:

- Enhanced Broadcasting.
- Interactive Broadcasting.
- Enhanced Packaged Media.
- Interactive Packaged Media.
- IPTV.
- OTT.

which can be extended with future additional profile definitions.

Clause 16 provides a registry of constants and clause 17 describes requirements for internet access clients.

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

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

NOTE 1: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

NOTE 2: The Java specifications which form the basis of the GEM specification are available at http://www.jcp.org. Additional information about Java TV is available at http://java.sun.com/javame/technology/javatv/index.jsp.

2.1 Normative references

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

[1] ETSI TS 102 727 (V1.1.1): "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.2.2".

NOTE: Availeble at http://www.etsi.org/deliver/etsi_ts/102700_102799/102727/.

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

[3] JSR 218: "Connected Device Configuration (CDC) 1.1".

NOTE: The latest release of Connected Device Configuration (CDC) is available at http://jcp.org/en/jsr/summary?id=218. At the time of writing, Connected Device Configuration (CDC) 1.1.2 is the latest maintenance release available at. It is strongly recommended to always use the latest release of this JSR.

[4] JSR 217: "Personal Basis Profile (PBP) 1.1".

NOTE: The latest release of Personal Basis Profile (PBP) is available at http://jcp.org/en/jsr/summary?id=217. At the time of writing, Personal Basis Profile (PBP) 1.1.2 is the latest maintenance release. It is strongly recommended to always use the latest release of this JSR.

[5] OCAP 1.1 Profile: "OpenCable Application Platform Specifications".

NOTE: Available at http://www.cablelabs.com/specifications/OC-SP-OCAP1.1.2-090930.pdf.

[6] ARIB STD-B23: "Application Execution Engine Platform for Digital Broadcasting," version 1.1, February 2004.

NOTE: Available at http://www.arib.or.jp/english/html/overview/sb_ej.html.

[7] ATSC A/101A: "Advanced Common Application platform (ACAP)", 12 February 2009

NOTE: Available at http://www.atsc.org/cms/standards/a_101a.pdf

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

[8] Open IPTV Forum - Release 2 Specification (October 7, 2010): "Volume 6 - Procedural Application Environment, V2.0".

NOTE: Available at http://www.openiptvforum.org/docs/Release2/OIPF-T1-R2-Specification-Volume-6-Procedural-Application-Environment-v2_0-2010-09-07.pdf

[9] ANSI/SCTE 90-1 2004: "SCTE Applications Platform Part 1 OCAP 1.0 Profile".

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

[10] Lindholm, T., Yellin, F. (Second Edition): "The Java Virtual Machine Specification".

NOTE: Available at http://java.sun.com/docs/books/jvms/.

[11] Gosling, J., Joy, B., Steele, G. (Third Edition): "The Java Language Specification".

NOTE: Available at http://java.sun.com/docs/books/jls/.

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

NOTE: http://tools.ietf.org/html/rfc3073.

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

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

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

NOTE: Available at http://www.etsi.org/deliver/etsi_en/300400_300499/300468/.

[15] ISO/IEC 10646-1: "Information technology - Universal Multiple-Octet Coded Character Set (UCS)".

[16] JSR 927: "Java TVTM API 1.1".

NOTE: The latest release of JavaTV is always available at http://www.jcp.org/en/jsr/detail?id=927. At the time of writing, JavaTV 1.1.1 is the latest maintenance release with no semantic changes against JavaTV 1.1. It is strongly recommended to always use the latest release of this JSR.

[17] DAVIC 1.4.1 Specification Part 9, Complete DAVIC Specifications, DAVIC. June 1999.

NOTE: Available at http://www.davic.org/Download/Spec1_4_1/141_part09.pdf.

[18] DVB - Digital Video Broadcasting - DVB Identifiers (website).

NOTE: Available at http://www.dvbservices.com/identifiers/index.php.

[19] ETSI ES 200 800: "Digital Video Broadcasting (DVB); Interaction channel for Cable TV distribution systems (CATV)".

[20] ETSI ETS 300 801: "Digital Video Broadcasting (DVB); Interaction channel through Public Switched Telecommunications Network (PSTN)/ Integrated Services Digital Networks (ISDN)".

[21] ISO/IEC 11172-2: "Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s -- Part 2: Video".

[22] ISO 6937: "Information technology -- Coded graphic character set for text communication -- Latin alphabet".

[23] ISO 8859 (all parts): "Information technology -- 8-bit single-byte coded graphic character sets".

[24] CIE 15-2004: Colorimetry, 3rd Edition - ISBN 978 3 901906 33 6.

[25] ISO 3166-1: "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

[26] IETF RFC 822: "Standard for the format of ARPA internet text messages".

[27] Hunt, R.W.G. 1987 "Measuring Colour (Ellis Horwood Series in Applied Science and Industrial Technology)". Ellis Harwood Limited, Chichester, England.

[28] W3C Recommendation 16 August 2006: "Namespaces in XML 1.0 (Second Edition)".

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

[29] ETSI TS 101 162: "Digital Video Broadcasting (DVB); Allocation of Service Information (SI) and Data Broadcasting Codes for Digital Video Broadcasting (DVB) systems".

[30] ETSI TS 101 225 (V1.1.1): "Digital Video Broadcasting (DVB); Home Local Network Specification based on IEEE 1394".

[31] CORBA/IIOP: "The Common Object Request Broker: Architecture and Specification", Object Management Group.

NOTE: Available at http://www.omg.org/cgi-bin/doc?formal/04-03-12.pdf.

[32] ETSI EN 301 192: "Digital Video Broadcasting (DVB); DVB specification for data broadcasting".

[33] GIF 89a: "Graphics Interchange Format (sm)", version 89a.

NOTE: Available at http://www.w3.org/Graphics/GIF/spec-gif89a.txt.

[34] ISO/IEC 10918-1: "Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines".

NOTE: Available at http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=18902.

[35] ISO/IEC 11172-3 (1993): "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3: Audio".

NOTE: Known as "MPEG-1 Audio".

[36] ISO/IEC 13818-1: "Information technology - Generic coding of moving pictures and associated audio information: Systems".

[37] ISO/IEC 13818-2: "Information technology - Generic coding of moving pictures and associated audio information: Video".

[38] ISO/IEC 13818-6 (1998): "Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extensions for DSM-CC".

[39] IEC 61966-2-1 (1999): "Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB".

[40] ITU-R Recommendation BT.470-6: "Conventional television systems".

[41] ITU-R Recommendation BT.709-5: "Parameter values for the HDTV standards for production and international programme exchange".

[42] JFIF: "JPEG File Interchange Format". Eric Hamilton, C-Cube Microsystems.

NOTE: Available at http://www.w3.org/Graphics/JPEG/jfif3.pdf.

[43] W3C Recommendation 1 October 1996, Version 1.0: "PNG (Portable Network Graphics) Specification".

NOTE: Available at http://www.w3.org/TR/REC-png-961001.html.

[44] IETF RFC 1321 (1992): "The MD5 Message-Digest Algorithm".

[45] IETF RFC 2616 (1999): "Hypertext Transfer Protocol -- HTTP/1.1".

[46] IETF RFC 2396 (1998): "Uniform Resource Identifiers (URI): Generic Syntax".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

[47] IETF RFC 768 (1980): "User Datagram Protocol".

[48] IETF RFC 791 (1981): "Internet protocol".

[49] IETF RFC 793 (1981): "Transmission control protocol".

[50] HAVi, v1.1, This comprises the following documents: HAVi v1.1 Chapter 8, 15-May-2001 HAVi v1.1 Java L2 APIs, 15-May-2001 HAVi v1.1 Chapter 7,15-May-2001.

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

[51] ITU-T Recommendation X.501 (1993): "Information Technology - Open Systems Interconnection - The Directory: Models".

[52] ITU-T Recommendation X.509 (1997): "Information technology - Open Systems Interconnection - The Directory: Authentication framework".

[53] IETF RFC 2313 (1998): "PKCS #1: RSA Encryption Version 1.5".

[54] ITU-T Recommendation X.680 (1994): "Information Technology - Abstract Syntax Notation One (ASN.1): Specification Of Basic Notation".

[55] IETF RFC 2459 (1999): "Internet X.509 Public Key Infrastructure Certificate and CRL Profile".

[56] ETSI ETS 300 706 (1997): "Enhanced Teletext Specification".

[57] FIPS PUB 180-1 (1995): "Secure Hash Standard".

NOTE: Available at http://www.itl.nist.gov/fipspubs/fip180-1.htm.

[58] IETF RFC 2246 (1999): "The TLS Protocol Version 1.0".

[59] IETF RFC 2045 (1996): "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

[60] W3C Recommendation 6 October 2000: "Extensible Markup Language (XML) 1.0 (Second Edition)".

NOTE: Available at http://www.w3.org/TR/2000/REC-xml-20001006.

[61] IETF RFC 1738 (1994): "Uniform Resource Locators (URL)".

[62] IETF RFC 1035 (1987): "Domain names - implementation and specification".

[63] Tiresias: "The RNIB/DTG "Tiresias" font version 8.03".

NOTE: For information on obtaining this font, please contact the DVB project office.

[64] IETF RFC 1034 (1987): "Domain names - concepts and facilities".

[65] IETF RFC 1982 (1996): "Serial Number Arithmetic".

[66] IETF RFC 2181 (1997): "Clarifications to the DNS Specification".

[67] ATSC A/100-5 (2003): " DTV Application Software Environment Level 1 (DASE-1) - Part 5: ZIP Archive Resource Format".

NOTE: Available at http://www.atsc.org/standards/a100/a_100_5.pdf.

[68] IETF RFC 1945 (1996): "Hypertext Transfer Protocol -- HTTP/1.0".

[69] W3C Recommendation 24 December 1999: "HTML 4.01 Specification".

NOTE: Available at http://www.w3.org/TR/html401.

[70] JSR 177 - SATSA, v1.0: "Security and Trust Services API (SATSA) for Java 2 Platform, Micro Edition".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 36 of 861

NOTE: Available at http://jcp.org/en/jsr/detail?id=177.

[71] IETF RFC 821 (1982): "Simple Mail Transfer Protocol".

[72] IETF RFC 977 (1982): "Network News Transfer Protocol".

NOTE: Available at http://www.w3.org/Protocols/rfc977/rfc977.

[73] IETF RFC 2818 (2000): "HTTP over TLS".

[74] IETF RFC 2368 (1998): "The mailto URL scheme".

[75] ISO/IEC 14496-18 (2004): "Information technology -- Coding of audio-visual objects -- Part 18: Font compression and streaming".

[76] JSR 172, v1.0: "J2ME Web Services Specification".

NOTE: Available at http://jcp.org/aboutJava/communityprocess/final/jsr172/index.html.

[77] JSR 216: "Personal Profile 1.1".

NOTE: Available at http://jcp.org/en/jsr/detail?id=216.

[78] JSR 219: FP 1.1: "Foundation Profile 1.1 for Java 2 Platform, Micro Edition".

NOTE: Available at http://jcp.org/en/jsr/detail?id=219.

[79] IETF RFC 3268 (2002): "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)".

[80] ETSI TS 102 034: "Digital Video Broadcasting (DVB); Transport of MPEG-2 Based DVB Services over IP Based Networks".

[81] ETSI TS 102 539: "Digital Video Broadcasting (DVB); Carriage of BCG information over IP".

[82] ETSI TS 102 816: "Digital Video Broadcasting (DVB); PVR/PDR Extension to the Multimedia Home Platform".

[83] IETF RFC 2326: "Real Time Streaming Protocol (RTSP)".

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

[85] ETSI, TS 102 817, “Digital Video Broadcasting (DVB);Digital Recording Extension to Globally Executable Multimedia Home Platform (GEM)” V1.1.1

[86] ETSI, TS 102 817, “Digital Video Broadcasting (DVB);Digital Recording Extension to Globally Executable Multimedia Home Platform (GEM)” V1.1.1

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area. For non-specific references, the latest version of the referenced document (including any amendments) applies.

[i.1] Application Definition Blu-ray Disc Format - BD-J Baseline Application and Logical Model Definition for BD-ROM, March 2005.

NOTE: http://www.blu-raydisc.com/ under “Technical Info”, “Public Specifications”

[i.2] ETSI EN 301 193 (V1.1.1): "Digital Video Broadcasting (DVB); Interaction channel through the Digital Enhanced Cordless Telecommunications (DECT)".

[i.3] ETSI EN 301 195 (V1.1.1): "Digital Video Broadcasting (DVB); Interaction channel through the Global System for Mobile communications (GSM)".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

[i.4] ETSI EN 301 199 (V1.2.1): "Digital Video Broadcasting (DVB); Interaction channel for Local Multi-point Distribution Systems (LMDS)".

[i.5] ETSI TS 101 211 (V1.9.1): "Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)".

[i.6] ETSI EN 300 472 (V1.2.2): "Digital Video Broadcasting (DVB); Specification for conveying ITU-R System B Teletext in DVB bitstreams".

[i.7] ETSI EN 300 743 (V1.2.1): "Digital Video Broadcasting (DVB); Subtitling systems".

[i.8] ETSI TR 101 201 (V1.1.1): "Digital Video Broadcasting (DVB); Interaction channel for Satellite Master Antenna TV (SMATV) distribution systems; Guidelines for versions based on satellite and coaxial sections".

[i.9] ETSI TR 101 202 (V1.1.1): "Digital Video Broadcasting (DVB); Implementation guidelines for Data Broadcasting".

[i.10] ETSI SR 001 262 (V1.7.1): "ETSI drafting rules".

[i.11] ETSI EN 301 790 (V1.2.2): "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems".

[i.12] IETF RFC 1877 (1995): "PPP Internet Protocol Control Protocol Extensions for Name Server Addresses".

[i.13] IETF RFC 1332 (1992): "The PPP Internet Protocol Control Protocol (IPCP)".

[i.14] IETF RFC 1661 (1994): "The Point-to-Point Protocol (PPP)".

[i.15] IETF RFC 1717 (1994): "The PPP Multilink Protocol (MP)".

[i.16] CENELEC EN 50049-1: "Domestic and similar electronic equipment interconnection requirements: Peritelevision connector".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 102 034 [80] and the following terms and definitions apply:

Application Program Interface (API): interface between an application and a particular feature, function or resource of GEM

application: functional implementation realized as software running in one or spread over several interplaying hardware entities

application boundary: concise general description of the data elements (HTML documents, code files, images, etc.) used to form one application and the logical locator of the entry point, the application boundary is described by a regular expression over the URL language

NOTE: Where no such boundary is drawn, the default boundary is the entire set of documents that the MHP platform can access.

application instance: unique invocation of an application, i.e. running the same application twice results in two distinct application instances

application manager: entity in GEM that is responsible for managing the lifecycle of the applications in the GEM terminal

NOTE: The application manager manages both the applications and non-applications.

autostart applications: this term has different definitions depending on the application format:

- A DVB-J autostart application is an application that is automatically loaded and executed by the Application Manager as soon as the user selects a service on which the application is signalled as autostart.
- Auto start application a application in a broadcast stream can be signalled as auto start in the same way that other DVB applications can.

NOTE: That it may not actually start providing service until it receives a start trigger.

best effort: implementation dependent approximation which is as close as reasonable in the circumstances concerned to what has been requested

character: specific "letter" or other identifiable symbol, e.g. "A"

character encoding: mapping between an integer input value, and the textual character that is represented by this mapping, e.g. in ASCII value 65 (decimal) is character "A", or shift-JIS for Japanese characters

character set: See character encoding.

communications network: system of interconnected entities providing data interchange between points or from a point to multiple points

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

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

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

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 39 of 861

definite: resource is considered of indefinite duration if the exact end point is not known in advance of accessing the resource (e.g. Transport Streams in MPEG-2)

NOTE: A resource is considered of definite duration if in principle the exact length is known (e.g. a file resource).

delegated application: application encoded in a representation that is not directly consumable by an MHP terminal's standard mechanisms

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

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

domain of an application: domain of an Xlet characterizes the "space" within which the Xlet is able to execute

NOTE 1: This includes both the "connection" where the Xlet is delivered and other "connections" where an already executing Xlet is allowed to continue executing.

NOTE 2: An application cannot run outside its domain. The maximum lifetime of an application extends from the moment the user navigates to its domain until the moment that the user navigates away from its domain.

NOTE 3: In the broadcast case a "connection" corresponds to a DVB-service. Broadcast signalling indicates which services can load an application and which services allow an already active application to continue.

DVB network: collection of MPEG-2 Transport Stream multiplexes transmitted on a single delivery system, e.g. all digital channels on a specific cable system

DVB-J: Java platform defined as part of the MHP specification

DVB-J API: one of the Java APIs standardized as part of the MHP specification

DVB-J application: set of DVB-J classes that operate together and need to be signalled as a single instance to the Application Manager so that it is aware of its existence and can control its lifetime through a lifecycle interface

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

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

events: asynchronous communication between applications and the MHP on which they are being executed, they provide communication between solution elements

font: mechanism that allows the specific rendering of a particular character to be specified - e.g. Tiresias, 12 point

NOTE: In practice a font file format will incorporate some aspects of a character encoding.

function: process which conveys or transforms data in a predictable way, it may be effected by hardware, software of a combination of the two

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

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

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

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

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

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 40 of 861

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

hardware entity: independent piece of hardware which forms part of a (multiple) local cluster of elements which as a whole is called a terminal

NOTE: A hardware entity is for example: a set top box, a digital VCR or a conditional access module. A hardware entity includes a number of resources. Each resource provides a number of functions.

hybrid: A combination of more than one GEM target in a terminal.

NOTE: Examples of hybrid GEM terminals include a hybrid Broadcast/Broadband terminal, a hybrid Broadcast/Internet terminal, other combinations are also possible.

indefinite: See definite.

interoperable plug-in: plug-in that only requires standard GEM APIs

interoperability: reception and presentation of applications in a vendor, author and broadcaster neutral framework

java API: standard interface for use by platform independent application software expressed in the Java language

java TV: APIs defined in the packages under the javax.tv. namespace in the present document

lifetime of an application: characterizes the time from which the application is Loaded to the time the application is Destroyed

locator: opaque reference to the location information of objects which are addressable within the Java TV API

NOTE: A given locator may represent a transport independent object and have multiple mappings to transport dependent locators.

Multimedia Home Platform (MHP): GEM-based terminal specification

native signalling: signalling infrastructure for a delegated application that are not part of a GEM terminal's standard mechanisms

navigator: resident application, typically provided by the manufacturer, which the end-user can activate at any time

NOTE: The navigator can be used to select services, applications, and initiate Interoperable applications.

object carousel: repetitively broadcast file system

OID: X.509 Object Identifier

OTT: OTT is a general term for video services delivered over the Open Internet. It's referred to as "over-the-top" because these services ride on top of plain Internet access service and don't require any business or technology affiliations with the network operator.

persistent storage: non-volatile memory available in the GEM terminal which can be read/written to by an application and which may outlive the application's own life

plug-in: set of functionality which can be added to a generic platform in order to provide interpretation of DVB registered, but non-DVB-J, application formats; e.g. HTML3.2 or MHEG-5

privileged application: application from a broadcaster, operator or service provider with permissions not granted to normal signed GEM applications

NOTE: This is the same concept as applications with MonitorAppPermission in OCAP [5].

profile: description of a series of minimum configurations, defined as part of the present document, providing different capabilities of the GEM terminal

NOTE: A profile maps a set of functions which characterize the scope of service options. The number of profiles is small. The mapping of functions into resources and subsequently into hardware entities is out of the scope of the present document and is left to manufacturers.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 41 of 861

PSI-only service: MPEG program listed in the PAT of a transport stream but not listed in the SDT or EIT

regular expression: method of capturing a large, possibly infinite set of strings in a compact representation

resident application: application available from non-volatile storage in a GEM device which may be expressed in DVB-J but need not be so

NOTE: Its delivery route is not specified.

resource: is a well defined capability or asset of a system entity, which can be used to contribute to the realization of a service

NOTE: Examples of resources include: MPEG decoder, Graphics system.

return channel: communications mechanism which provides connection between GEM and a remote server

running application: application is considered running if it is in any state other than NOT_LOADED or INVALID

runtime code extension: ability to extend, at runtime, the executable code of an application with code not originally packaged with the application

NOTE: This facility is provided, for example, by org.dvb.lang.DVBClassLoader in DVB-J and by eval() in ECMAscript.

sandbox: unsigned applications and signed applications without a permission file have access to all the APIs for which there is no permission signalling defined, this is commonly called the sandbox

service: sequence of programs under the control of a broadcaster which can be broadcast as part of a schedule

service component: part of a service

NOTE: For a DVB service, service components are normally the MPEG elementary streams listed in the PMT for the service. Where multiple streams of subtitles or MPEG-2 audio representing different languages are carried in the same MPEG elementary stream these are logically service components but are not exposed through the GEM APIs as being distinct and separate.

stand-alone application: application which is not related to a conventional DVB service

NOTE: In this version of the present document, this includes applications running as part of a stored service and applications whose signalling was carried over the interaction channel.

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

stream: unidirectional continuous flow of content, for example, MPEG2 video

stored service: set of GEM applications stored within the GEM terminal and treated as a service with regard to life cycle, service selection, etc.

subsidizing service provider: operator who is subsidizing all or part of the cost of a GEM terminal and hence expects special privileges for their applications

system software: software implementation below the API for a specific platform entirely under control of the manufacturer

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

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

transport dependent: transport dependent Service or Locator refers to a service or content delivered by a particular delivery mechanism or channel

EXAMPLE: BBC1 delivered by DVB-T would have a different transport dependent Service and Locator from BBC1 delivered by IPTV.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 42 of 861

transport independent: transport independent Service or Locator can refer to copies of that service or content regardless of how they are delivered to the GEM terminal

EXAMPLE: The same transport independent Service and Locator could apply to both BBC1 delivered by DVB-T and BBC1 delivered by IPTV.

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

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

tuning: act of switching between two MPEG transport streams or multiplexes

NOTE: Switching between two DVB services carried in the same transport stream is not tuning.

viewer: end-user of the GEM terminal

xlet: interface used for DVB-J application life cycle control

The following definitions from clause 3.2.1 "Definitions" of OCAP [5] shall apply:

- Abstract service.
- Unbound application.

3.2 Abbreviations

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

AIT	Application Information Table
API	Application Programming Interface
AV	Audio Video
AWT	Abstract Windowing Toolkit
BCG	Broadband Content Guide
CA	Conditional Access
CI	Common Interface
CLUT	Colour LookUp Table
CoD	Content on Demand
CRID	Content Reference Identifier
CSS	Cascading Style Sheets
DAVIC	Digital Audio Visual Council
DCT	Discrete Cosine Transformation
DECT	Digital Enhanced Cordless Telecommunications
DN	Distinguished Name
DOM	Document Object Model
DSM-CC	Digital Storage Media - Command and Control
DVB	Digital Video Broadcasting
DVB-J	DVB Java
EPG	Electronic Program Guide
ETSI	European Telecommunications Standards Institute
GEM	Globally Executable MHP
GIF	Graphics Interchange Format
GSM	Global System for Mobile communications
GUI	Graphical User Interface
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transport Protocol
I/O	Input / Output
ICC	International Colour Consortium
ID	IDentifier
IHDN	In Home Digital Network
IP	Internet Protocol
IPR	Intellectual Property Rights

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

IPTV	IP television
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JDK	Java Development Kit
JFIF	JPEG File Interchange Format
JMF	Java Media Framework
JPEG	Joint Picture Expert Group
LMB	Live Media Broadcast
LMDS	Local Multipoint Distribution System
MHEG	Multimedia Hypermedia Expert Group
MHP	Multimedia Home Platform
MPEG	Moving Picture Expert Group
MSO	Multiple System Operator
NPT	Normal Play Time
OC	Object Carousel
OCAP	Open Cable Application Platform
OOB	Out of Band
OS	Operating System
OSD	On Screen Display
OTT	Over the Top TV
PFR	Portable Font Resource
PMT	Program Map Table
PNG	Portable Network Graphics
POD	Point of Deployment
PSI	Program Specific Information
PSTN	Public Switched Telephone Network
PVR	Personal Video Recorder
RAM	Random Access Memory
RGB	Red, Green Blue
SCART	The connector known as "Syndicat des Constructeurs d'Appareils Radiorecepteurs et Televiseurs" also Peritel and Euroconnector and Cenelec EN 50049-1 [i.16].
SD&S	Service Discovery and Selection
SI	Service Information
SMATV	Satellite Master-Antenna Television
TCP	Transmission Control Protocol
TS	Transport Stream
UCS	Universal Multiple-Octet Coded Character Set
UDP	User Datagram Protocol
UI	User Interface
URL	Uniform Resource Locator
UTF	UCS Transformation Coding
UTF8	Universal Transformation Format 8
UU	User to User
VM	Virtual Machine
WSS	Wide Screen Signalling
XML	eXtensible Markup Language

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

4 General considerations and conventions

4.1 General considerations

4.1.1 Purpose

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

A GEM-based specification, the Multimedia Home Platform (MHP) middleware standard, defines a comprehensive platform that enables interactive television services to be deployed that are interoperable across any manufacturer's implementations of the standard. MHP is a comprehensive specification of a receiving device (an MHP terminal). MHP terminals receive digital video broadcasting services based on standards for various transmission media including satellite, cable, terrestrial, microwave and TCP/IP. The transport layer may be DVB-T/T2, DVB-C/C2, DVB-S/S2 or an IP transport.

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 compatible network.

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

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

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

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

NOTE: The present document covers the broadcast, packaged media, IPTV and OTT targets and defines the common core which is mandatory. There are a few APIs that are not required when only the broadcast target is implemented. Similarly, there are a few APIs that are not required when only the IPTV, OTT or packaged media target is implemented. However, when one or both targets are implemented, it is mandatory that all the required APIs and other definitions are included as specified in this recommendation.

4.1.2 Format

The present document does not state how the receiver has to be built or what network infrastructure has to underlie the implementation; it is limited to specifying the behaviour and interfaces that globally interoperable applications may rely on.

4.1.3 Inclusion of GEM features

4.1.3.1 Subsetting prohibited

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

4.1.3.2 Supersetting permitted

If a GEM terminal specification wishes to include APIs, signalling or behaviours defined in MHP [1] that are not required by GEM, it may do so as described in clause 15.6, "Functional equivalents".

4.1.4 Addition of non-GEM interfaces

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 [5] defines extensions in the Java package `org.ocap`.

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

4.1.4.1 DVB-J enumerations

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

NOTE: For example, the interface `org.dvb.net.rc.RCInterface` defined in annex R introduces an enumeration that is returned by the method `getType()`. This enumeration includes the values `TYPE_CATV`, `TYPE_DECT`, etc. It is not permissible to attempt to subdivide one of these types by introducing new enumeration values in a different namespace. See also the example in clause W.2, "Example of enumeration extension".

4.1.4.2 Competitive APIs

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

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

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

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

4.1.4.2.1 Illustration of Complementary Functional equivalents

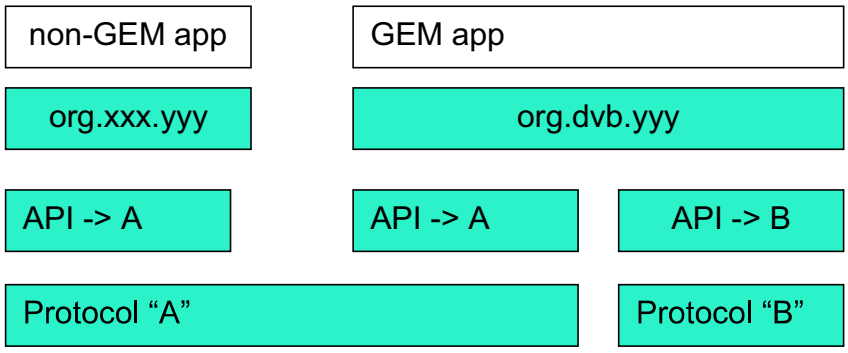


Figure 1: Complementary Functional equivalents

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

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

4.1.4.2.2 Illustration of Competitive APIs

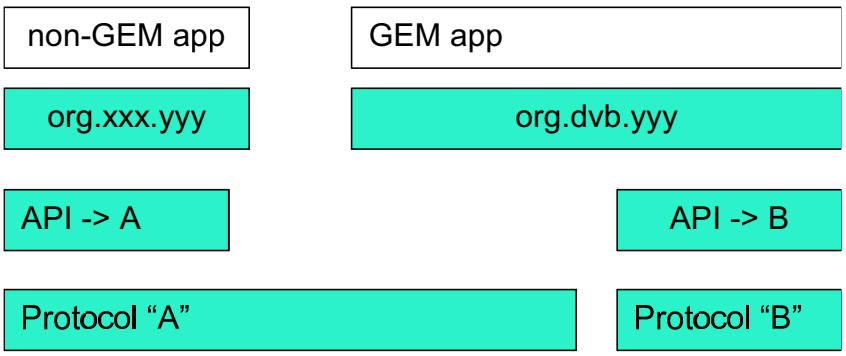


Figure 2: Competitive APIs

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

4.1.5 Application areas

GEM defines four targets: Broadcast, Packaged Media and IPTV and OTT. The first two of these, Broadcast and Packaged Media, are further subdivided into two application areas: Enhanced Broadcasting and Interactive Broadcasting. Enhanced Broadcasting combines digital broadcast of audio/video services with downloaded applications which can enable local interactivity; it does not need an interaction channel. Interactive Broadcasting enables a range of interactive services associated or independent from broadcast services; it requires an interaction channel.

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

Currently GEM defines four targets:

- The broadcast target, which is used by GEM terminal specifications in a broadcast environment, such as MHP [1], OCAP [5], ATSC [7] and ARIB [6].
- The packaged media target where the media is packaged on a physical medium which is possibly read-only, such as an optical disc as Blu-ray [13].
- The IPTV target, where media is transmitted over a bidirectional broadband connection, such as in MHP [1] and in the OpenIPTV Forum [8].
- The OTT target, where media is transmitted over a bidirectional broadband connection without QoS guarantee.

Combinations of the GEM targets into a combined target is explicitly permitted, following the GEM rules for combined profiles as defined below:

- If a feature is present in one of the targets, it shall be present in the combined target.
- The combined target is the unification of all features of the individual target.
- If a feature is optional or not required in the combined target, a valid implementation may omit it.
- If a feature is optional in one target and mandatory in the other, it shall be mandatory in the combined target.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

4.1.5.1 Broadcast target

The broadcast target is available since GEM 1.0.x and is defined for broadcast TV using cable, terrestrial or satellite.

4.1.5.2 Packaged Media target

The packaged media target allows to combine audio/video content and application into a self contained medium, such as an optical disc. The GEM packaged media target is the basis of the BD-J Blu-ray execution environment.

4.1.5.3 IPTV target

The GEM-IPTV target is defined to enable audio/video services over managed broadband networks. These networks allow management of the broadband bandwith and can maintain a guaranteed QoS.

Since these networks can provide both unicast and multicast network services, typical services can include LMB (based on multicast) and unicast Content on Demand (CoD) services.

4.1.5.4 OTT target

According to the service profiles defined in [80] for delivery of media contents over IP networks, OTT is restricted to Content on Demand (CoD) type of services and LMB over unicast connections.

NOTE: Multicast based LMB is typically not possible within OTT, even if live (unicast) contents can be enjoyed via OTT.

Content on Demand (CoD) service is a service where users can select the individual content items they want to watch from the list of available content. Consumption of the content is started upon user request.

2 types of CoD services are addressed within OTT target:

- Streamed CoD services, where content is consumed while the content itself is being delivered (real-time streaming)
- Download CoD services, where the whole content has to be downloaded first to the local storage in the receiver before consuming it. Consumption is then independent of the delivery.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

4.1.5.5. Typical Hybrid profiles

The following combined targets are most useful for certain markts:

- OTT+Broadcast: a combiation of OTT with the Broadcast Interactive Profile used in hybrid broadcast/broadband receivers
- OTT+Packaged: a combination of OTT with the Packaged Media Interactive Profile
- OTT+IPTV: a combined IP profile for use in managed and unmanaged networks
- OTT+Broadcast+Packaged: a combination of OTT with the Broadcast Interactive Profile with Packaged Media Interactive Profile, in all-in-one hybrid broadcast/broadband receivers/players

4.1.6 Profiles

As not all GEM implementations 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, yielding five profiles.

Figure 3 shows the six GEM profiles. These are described in greater detail in clause 15, "Detailed platform profile definitions".

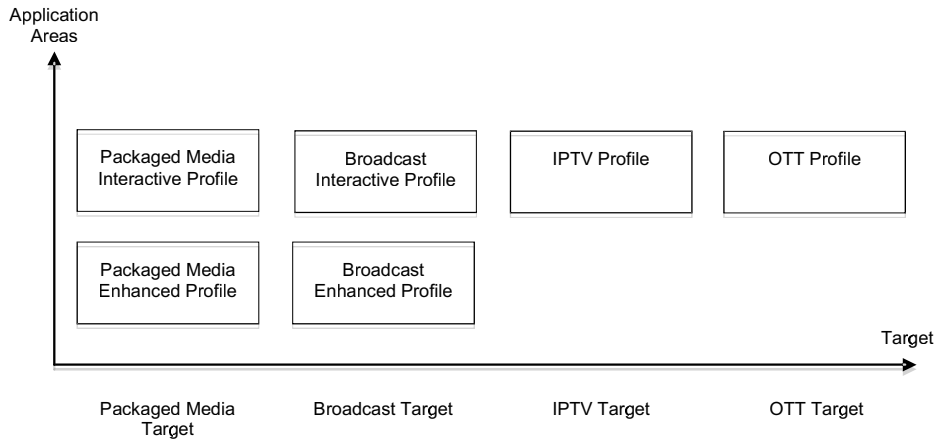


Figure 3: GEM platform profiles

4.1.7 Full conformance with the present document

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

Further, GEM terminals shall be conformant with any one of the following specifications:

- MHP 1.0., MHP 1.1, MHP 1.2, MHP 1.2.2. [1] or later.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- OCAP 1.0, OCAP 1.1 [5] or later.
- ARIB STD-B23[6]: "Application Execution Engine Platform for Digital Broadcasting," version 1.1, February 2004 or later.
- ATSC A/101[7]: "Advanced Common Application platform (ACAP)" or later.
- SCTE 90-1 [9].
- Blu-Ray Disc Player Specification [13].
- PAE 1.0 specification of the Open IPTV forum [8] or later.
- Any other specification that implements the IPTV target, for the purposes of presenting GEM services delivered via an IP transport. It is permissible if such an IPTV specification is not publicly available

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

- Any other specification that implements the IPTV target, for the purposes of presenting GEM services delivered via an IP transport. It is permissible if such an IPTV specification is not publicly available.

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

NOTE 1: It is expected that a future version of OCAP will adopt GEM 1.2.x.

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

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

"7.1 Compliance with GEM

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

- Arch
- Carousel
- Text Wrapping

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

7.1.1 GEM errata

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

4.2 Conventions

Unless otherwise specified, the BNF notation in the present document shall follow the definitions of section 2.1 of RFC 2616 [45].

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 50 of 861

The text "Void" in a clause title or in a reference designates a clause or reference that existed in a previous version of the present document that has been removed and replaced with "Void" to preserve the numbering of the text that remains.

The inclusion of a normative reference does not imply the inclusion of all of that document. The clauses of a normative reference that apply to the present document are identified where each normative reference is used.

The javadoc accompanying the present document shows the maximum set of APIs and their dependencies. Not all of these are mandatory in all GEM terminals. (For example, org.dvb.service.DvbSIManager is shown as implementing CRIDAccess and SDSRecordAccess even though CRIDAccess is defined in clause 11.6.7, "Integration between protocol independent SI API and TV-Anytime" and hence is only required when the broadband content guide is supported.) Implementations not including an API should omit the reference to it. The present document has been designed to ensure that these omissions happen at clean break points.

4.2.1 Void

4.2.2 Void

4.2.3 Void

4.2.4 Conventions within the present document

4.2.4.1 GEM

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

4.2.4.2 Resident navigator

The present document uses the terms "navigator" and "resident navigator". It is noted that in GEM terminal specifications, it is permissible for some of the functions of the navigator to be delegated to an entity that is not part of the resident software of the terminal, e.g. the OCAP [5] monitor application.

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

4.2.4.3 DVB service

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

4.2.5 References to OCAP

Where the words "shall be supported" or "may be supported" are used referring to a clause from OCAP [5], the following terms in the referenced clause shall be re-defined as follows:

- References to OCAP-J applications or application instances shall be interpreted as references to GEM applications.
- In the case of broadband networks, references to the following XAIT descriptors shall be interpreted as references to the descriptors of the same name defined in clause 10.16.2 "XAIT" of MHP [1].
    - Abstract Service Descriptor.
    - Unbound Application Descriptor.
    - Privileged Certificate Descriptor.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- Requirements on and references to manufacturer applications or host device manufacturer applications or receiver manufacturer applications shall be ignored as they are outside the scope of the present document.
- References to MSO shall be interpreted as references to network operators in general.
- References to the monitor application or monitor applications shall be interpreted as references to privileged applications.
- References to OCAP implementations shall be interpreted as references to GEM implementations.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

5 Basic architecture (informative)

GEM does not mandate a basic architecture. The section below describes an informative example of one possible architecture for GEM terminal specifications.

5.1 Context

At its simplest level, GEM is set in the following context (see figure 4). The software of GEM has access to flows of streams and data, and may write some data to storage. The platform may be able to route streams and data outwards to a sink or store.

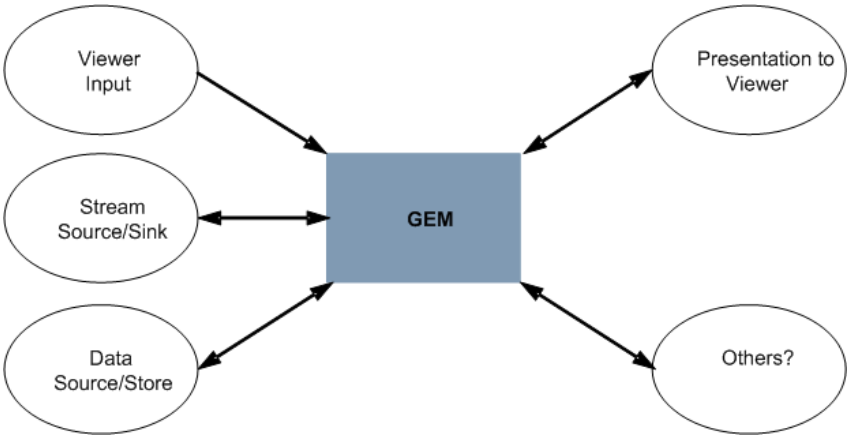


Figure 4: GEM context

The platform will receive inputs from Viewer input devices and output communications through a screen or other outputs like loudspeakers to present to the viewer. The platform may have access to communications with remote entities.

The diagram in figure 5 shows a possible set of external interfaces between a GEM terminal and the outside world. This is one example only but it serves to illustrate a series of principles.

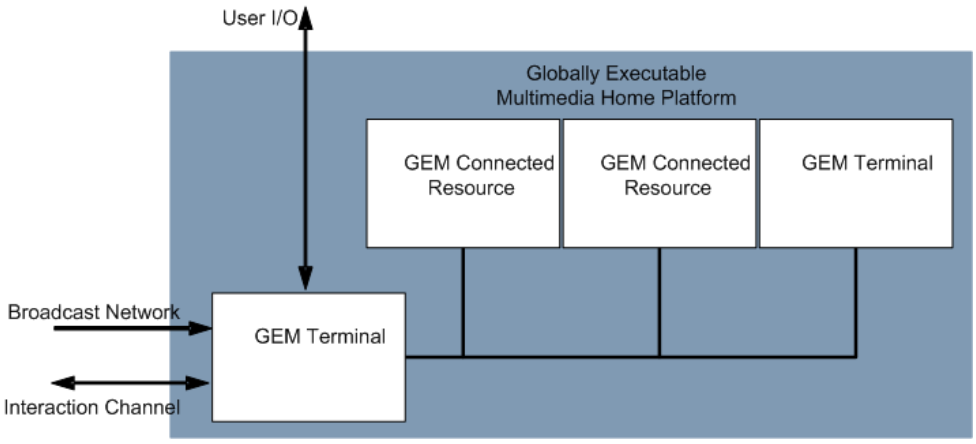


Figure 5: External interfaces between a GEM terminal and the outside world

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The resources of the GEM terminal, accessible by an application, may be contained in a series of different but connected physical entities.

The local cluster may connect a number of GEM terminals and resources.

A cluster may also include resources which are not part of the GEM infrastructure and are not available to the application.

The local cluster is understood to be consistent with the DVB IHDN specification [30]. The detailed description of GEM in the local cluster is not in the first version of the specification.

5.2 Architecture

The Architecture describes how the GEM software elements are organized.

The GEM model considers 3 layers (figure 6):

- Resources.
- System software.
- Applications.

The API lies between the Applications and the System Software seen from the perspective of an application.

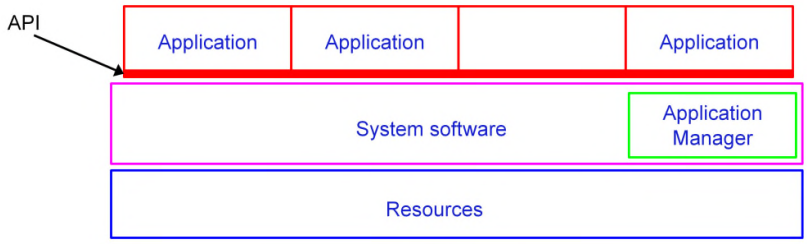


Figure 6: Basic architecture

5.2.1 Resources

The hardware entities in the platform include a number of functions. They are represented by hardware or software resources. There is no assumption about how they are grouped. The model considers that there can be more than one hardware entity in the total Platform.

From an abstract point of view it makes no difference if the logical resources are mapped into one or several hardware entities. What is important is that resources are provided to the GEM terminal transparently. An application should be able to access all locally connected resources as if they were elements of a single entity.

5.2.2 System software

Applications will not directly address resources. The system software brings an abstract view of such resources. This middle layer isolates the application from the hardware, enabling portability of the application.

The implementations of the Resources and System software are not specified in the present document.

5.2.2.1 Application Manager

The system software includes an application management function, which is responsible for managing the lifecycle of all applications, including Interoperable ones.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

5.2.3 Application

Applications implement interactive services as software running in one or more hardware entities. The interface for GEM applications is a top view from application to the system software.

Figure 6 illustrates an idealized architecture model of the processes which will occur in a GEM terminal. A hierarchy of control is assumed in which each layer controls the processes in adjacent layers. The top layer is responsible for the control of the operation via interactive applications. The Application Manager is part of the System Software and as such is implementation specific. It interacts with all applications.

The System Software implements the API by presenting an abstract model of:

- Streams played from different sources and pipes for conducting them.
- Commands and events.
- Data records or files.
- The hardware resources.

The API provides the associated services to applications.

In fact there are many APIs which implement distinct services and interfaces. These are described in detail in the present document, either by reference to external documents, or by detailed specification.

The present document describes the interfaces between the network, the application and the system software of the GEM terminal. The implementation of any of these three is not specified in the present document.

5.3 Interfaces Between a GEM Application and the GEM Terminal

Application(s) use the API to access the actual resources of the receiver, including: databases, streamed media decoders, static content decoders and communications. These resources are functional entities of the receiver and may be finally mapped onto the hardware of the receiver in some manner.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

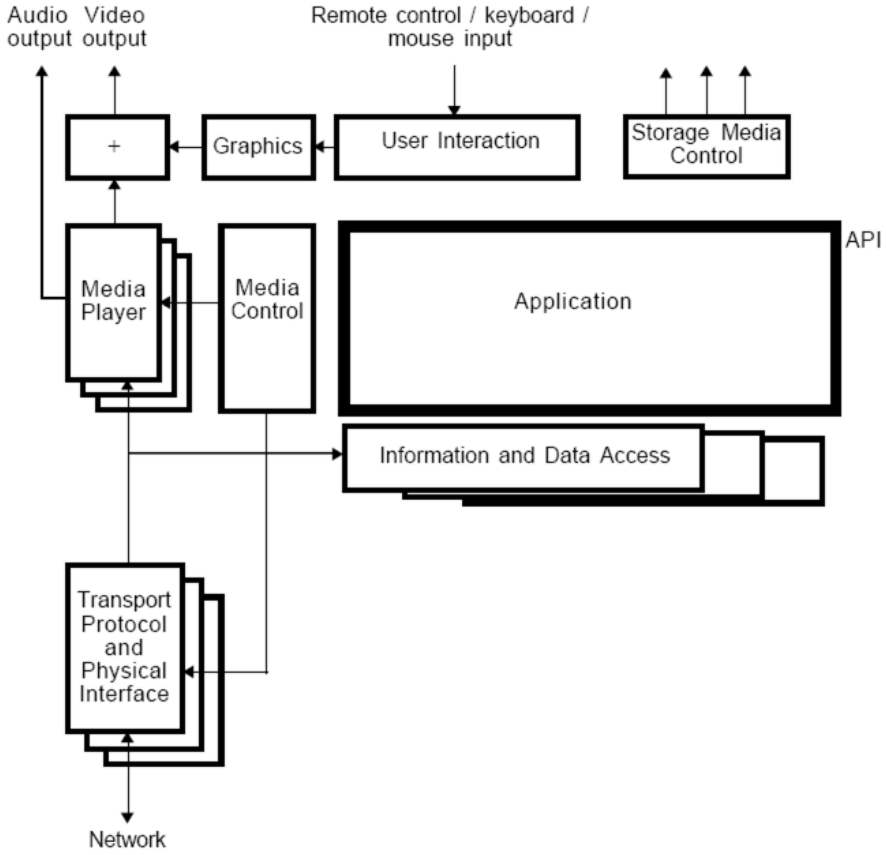


Figure 7: Interfaces between a GEM application and the GEM system

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

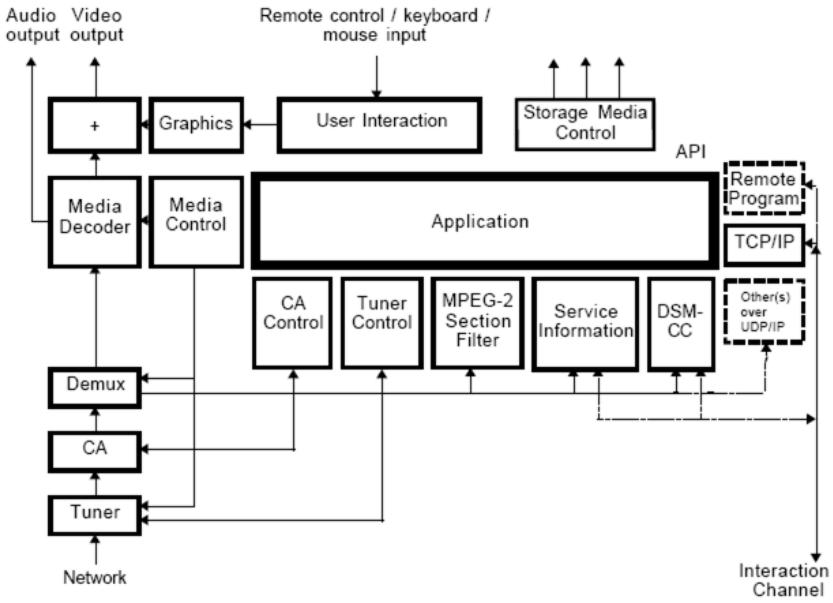


Figure 8: Interfaces between a GEM application and the GEM platform

The diagrams in figures 7 and 8 show these interfaces and their relationships to media and information flows within a GEM system - excluding any local cluster issues. The first diagram shows a generic GEM, the second a specific instance of an enhanced broadcast or interactive TV profile system, with optional additions shown.

In figures 7 and 8, only the border between the application and the rest of the system is in the scope of the present document. All the rest of each diagram is implementation dependent and shown for information purposes only.

5.4 Plug-ins

A "plug-in" is a set of functionality that can be added to a generic platform in order to provide interpretation of application and content formats not specified by the present document to be included in GEM terminals.

NOTE: Those organizations concerned with interoperation between the standard GEM platform and other platforms need to specify the plug-in properly for such platforms.

The choice of which plug-ins to use must be in the hands of the end-user in order that he can have a choice of sources of service. This option can be exercised in a number of ways, including the purchase of equipment with "built-in" plug-in functionality, the positive selection of a download, or the automatic selection of a download where there is no memory resource limitation.

The plug-in may stay resident where the design of the platform implementation allows. The GEM terminal including the plug-in must behave, once the plug-in is loaded and operational, in the same way as a platform supporting the format of the delegated applications without the use of a plug-in.

Figure 9 illustrates the position of two types of Plug-in in the GEM terminal.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

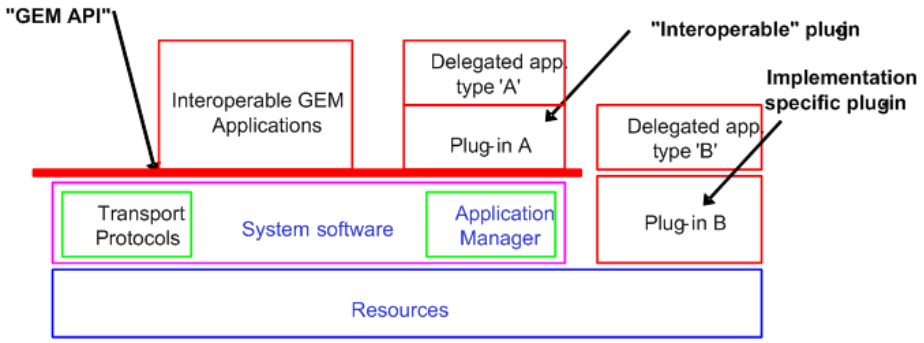


Figure 9: Illustrative plug-in implementation options

There are two possible types of Plug-in implementation:

- Using implementation-specific code (e.g. in native code, or using implementation-specific Java APIs). This is called an implementation- specific plug-in, illustrated as plug-in "B" in the figure.
- A GEM application. This is called an interoperable plug-in, illustrated as plug-in "A" in the figure.

The internal specification of both plug-ins ("A" and "B") is outside of the scope of the present document. They are illustrated to show their relationship to the platform.

There are two varieties of inter-operable plug-ins depending on the signalling used for the delegated applications.

- Ones which work with delegated applications signalled with GEM signalling and which implement `org.dvb.application.plugins.Plugin`. These delegated applications are visible to the GEM receiver and are managed by the GEM application manager.
- Ones which work with delegated applications signalled with native signalling and only look like a GEM application to the GEM receiver. These do not implement the plug-in interface. These delegated applications are invisible to the GEM receiver and totally managed by the plug-in.

5.4.1 Security Model

Plug-ins must have sufficient access to the resources in the platform to implement the specification concerned. An implementation-specific plug-in may have access to many of the resources of the platform irrespective of the GEM security model. All plug-ins are responsible for managing the security of the applications they execute.

If a plug-ins needs privileged access to resources not available to all downloaded applications (i.e. not in the "sand box") in order to provide equivalent function to the "legacy" supported they will require the appropriate authentication.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

6 Transport protocols

6.1 Introduction

For broadcast targets, in order to be able to talk to the external world, a GEM terminal has to communicate through different network types.

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

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

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

6.2 Broadcast channel protocols

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

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

Figure 10 shows the broadcast channel protocol stack for GEM. As some of the protocols are not required by the present document, not all elements of this figure necessarily apply. See clause 15.6, "Functional equivalents".

The full details of APIs that provide DVB-J applications with access to broadcast protocols are in clause 11, "DVB-J platform".

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

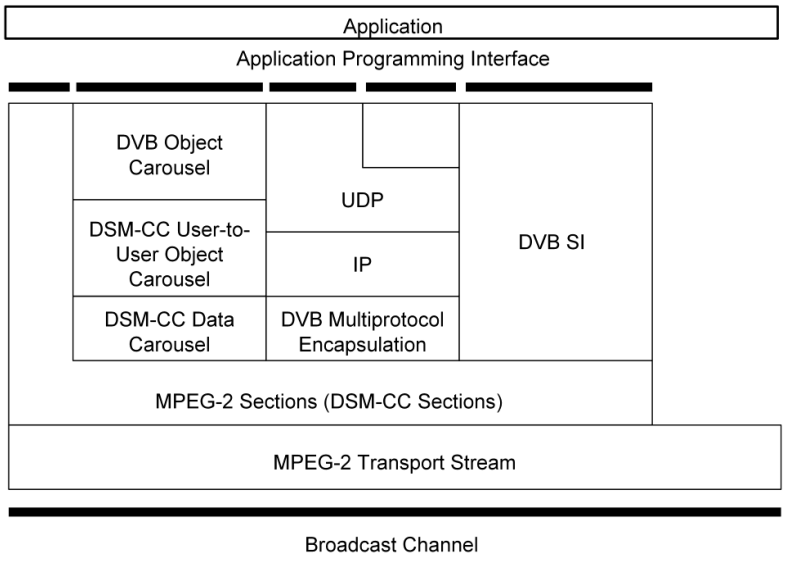


Figure 10: Broadcast Channel Protocol Stack

Except in the case of MPEG-2 sections (see clause 6.2.2, "MPEG-2 sections"), when a GEM application attempts to access conditional access scrambled data through one of these broadcast channel protocols, the GEM terminal shall attempt to initiate descrambling of this data without the application needing to explicitly ask for it. Attempts to access

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

conditional access scrambled data at the level of MPEG-2 sections shall not happen without the application explicitly asking for this. This requirement applies to GEM terminal specifications that include the functional equivalent named "Conditional Access" as specified in clause 15.6, "Functional equivalents".

6.2.1 MPEG-2 transport stream

MPEG-2 Transport Stream is defined in ISO/IEC 13818-1 [36].

6.2.2 MPEG-2 sections

MPEG-2 private sections as defined in ISO/IEC 13818-1 [36] is based on the MPEG-2 Transport Stream protocol in clause 6.2.1.

6.2.3 DSM-CC private data

DSM-CC Private Data protocol as defined in ISO/IEC 13818-6 [38].

6.2.4 DSM-CC data carousel

DSM-CC Data Carousel as defined in ISO/IEC 13818-6 [38].

6.2.5 Object carousel

DSM-CC User-to-User Object Carousel protocols as defined in ISO/IEC 13818-6 [38] with the restrictions and extensions as defined in EN 301 192 [32], TR 101 202 [i.9] and annex B, "Broadcast filesystem and trigger transport".

Use of the Object Carousel protocol is not required for GEM terminal specifications. GEM terminal specifications that do not include the functional equivalent named "Object Carousel" as specified in clause 15.6, "Functional equivalents" shall specify a functional equivalent that satisfies the requirements of the API specified in Annex P, "Broadcast transport protocol access".

6.2.5.1 Void

6.2.5.2 Void

6.2.5.3 Loss of carousel behaviour

Under some conditions, carousel data streams servicing broadcast file systems may become unavailable. The conditions for temporary disconnection and reconnection of carousel are defined in clause 9.1.5, "Persistence of Applications Across Service Boundaries".

DSM-CC User-to-User Object Carousels as described in clause 6.2.5, "Object carousel" become permanently unavailable due to changes in the signalling including the following:

- The component signalled as carrying the DSI is removed from the PMT.
- The value of carousel ID associated with the carousel changes.
- The program disappears from the PAT.
- After an implementation dependent time general failure of the signalling (e.g. non-transmission of the PMT).

Additionally, carousels also become permanently unavailable when loss of connection to a temporarily disconnected carousel becomes permanent.

The conditions for permanent loss of a carousel may be specified differently in GEM terminal specifications that do not include the DSM-CC User-to-User Object Carousel as described in clause 6.2.5, "Object carousel". However, GEM terminal specifications shall specify conditions for permanent loss of a carousel. The conditions for temporary

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

disconnection and reconnection of a carousel defined in clause 9.1.5, "Persistence of Applications Across Service Boundaries" apply to all GEM terminal specifications.

When carousel data streams become unavailable, implementations may continue to provide data from carousels which have been lost where they have that data cached. The extent of this is clearly implementation dependent (but limited by clause B.5, "Caching"). It is also implementation dependent how this changes with time. Permanently lost carousels shall never be restored automatically. Temporary disconnections and reconnections are automatic and are largely invisible to the application. However, implementations must preserve Java IO semantics. For example, the `InputStream.available()` method should accurately report the number of bytes available without blocking.

Data not in such a cache shall be unavailable to applications. When applications attempt to access unavailable data from permanently lost carousels, the operation shall fail. The failure mode shall be one appropriate to the content format and the mechanism being used to access the data.

Failure modes for DVB-J applications are defined in clause 11.5.1.3, "Behaviour following loss of a broadcast carousel".

When an application attempts to access unavailable data from a temporarily disconnected carousel, the operation shall block until the data becomes available, or it is interrupted. This includes any operations that indirectly cause synchronous loading operations, as stated in clause 11.5.1.1, "Constraints on the java.io.File methods for broadcast carousels".

Upon reconnection to a carousel, the system shall start fetching any outstanding data. Any blocked I/O operations shall return once the data is received.

A temporarily disconnected service may become permanently lost if the system determines that the loss of connection is irrecoverable - as stated in annex P "(normative): Broadcast Transport Protocol Access", org.dvb.dsmcc.ServiceDomain. The cases in which this may occur are:

- The carousel becomes permanently unavailable, as stated above.
- The period since the carousel became disconnected is at least 60 seconds.

If this occurs, any blocked I/O operations shall terminate with `InterruptedIOException` and the `ServiceDomain` shall enter the detached state. An implementation may choose to use longer timeouts or other strategies to determine when to permanently lose carousels, within the constraints listed above. For example, boot carousels for running applications may be kept in preference to carousels mounted by applications.

6.2.6 Protocol for delivery of IP multicast over the broadcast channel

DVB Multiprotocol Encapsulation as defined in EN 301 192 [32] provides support for IP and is based on the DSM-CC Private Data protocol. EN 301 192 [32] only defines support for carriage of multicast IP in MPE and not support for carriage of unicast IP.

Use of the DVB Multiprotocol Encapsulation protocol is not required for GEM terminal specifications. If, however clause 11.5.2, "Support for Multicast IP over the Broadcast Channel" is supported, some mechanism for delivery and signalling of IP multicast over the broadcast channel shall be specified.

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

6.2.7 Internet Protocol (IP)

Internet Protocol as defined in RFC 791 [48].

6.2.8 User Datagram Protocol (UDP)

User Datagram Protocol as defined in RFC 768 [47].

6.2.9 Service information

Use of DVB service information is not required for GEM terminal specifications, however some mechanism for delivery of service information that is sufficient for the SI access mechanisms required by GEM shall be specified.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 61 of 861

DVB Service Information as defined in EN 300 468 [14] and ETR 211 [i.5] may be taken as an informative example of such a mechanism for GEM terminal specification that do not include the functional equivalent named "SI" as specified in clause 15.6, "Functional equivalents".

6.2.10 IP signalling

IP Notification Table (INT) as defined in EN 301 192 [32].

Use of this signalling is not required for GEM terminal specifications, however a functional equivalent that satisfies the requirements of clause 11.5.2, "Support for Multicast IP over the Broadcast Channel" is required if IP over the broadcast channel is supported.

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

6.3 Interaction channel protocols

This clause deals with the DVB defined or referenced interaction channel protocols. This clause does not consider other protocols and the APIs that would provide access to them. Other private protocols and possibly APIs are not precluded and are outside of the scope of GEM.

Figure 11 illustrates the set of DVB defined interaction channel protocols that are accessible by GEM applications in some or all profiles (see clause 15, "Detailed platform profile definitions"). The full details of the APIs that provide access to these interaction protocols are in clause 11, "DVB-J platform".

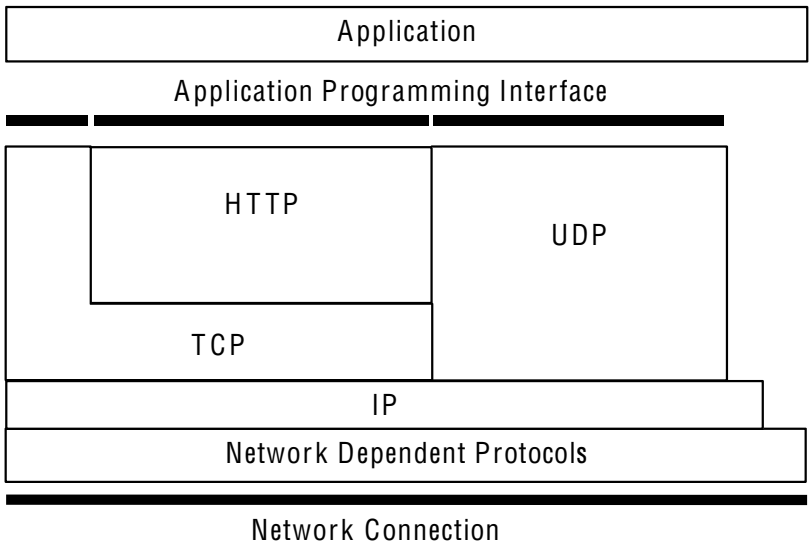


Figure 11: Interaction Channel Protocol Stack

6.3.1 Network Dependent Protocols

As defined in ES 200 800 [19], ETS 300 801 [20], EN 301 193 [i.2], EN 301 195 [i.3], EN 301 199 [i.4], TR 101 201 [i.8], EN 301 790 [i.11] respectively for CATV, PSTN/ISDN, DECT, GSM, LMDS SMATV and satellite networks.

For connections based interaction channels, the PPP protocol is used as defined in RFC 1332 [i.13], RFC 1661 [i.14], RFC 1717 [i.15]. Network supplied DNS server addresses shall be supported as in RFC 1877 [i.12].

NOTE: The protocols listed in this clause are not explicitly required in any GEM profile.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

6.3.2 Internet Protocol (IP)

Internet Protocol as defined in RFC 791 [48].

6.3.3 Transmission Control Protocol (TCP)

Transmission Control Protocol as defined in RFC 793 [49].

6.3.4 UNO-RPC

The UNO-RPC consists of the Internet Inter-ORB Protocol (IIOP) as specified in CORBA/IIOP [31].

6.3.5 UNO-CDR

The UNO-CDR as defined in CORBA/IIOP [31].

6.3.6 DSM-CC User to User

DSM-CC User-to-user as defined in ISO/IEC 13818-6 [38] with the restrictions and extensions as defined in EN 301 192 [32] and TR 101 202 [i.9].

6.3.7 Hypertext Transfer Protocol (HTTP)

6.3.7.1 HTTP 1.1

Hypertext Transfer Protocol as defined in RFC 2616 [45].

NOTE: HTTP 1.1 support as specified is not required in the broadcast, packaged media and IPTV target of GEM. This does not preclude a GEM terminal specification using a different profile of HTTP.

6.3.7.2 GEM profile of HTTP 1.0

HTTP 1.0 is defined in RFC 1945 [68]. GEM terminals supporting HTTP 1.0 are required to support persistent connections as defined below. GEM terminals implementing profiles where HTTP 1.0 is required are also allowed to implement subsequent versions of the HTTP specification (e.g. HTTP 1.1, RFC 2616 [45]) as long as these are backwards compatible and persistent connections or their equivalent in the subsequent version of the specification are supported.

With no support for persistent connections, a separate TCP connection has to be established to fetch each URL, increasing the load on HTTP servers and causing congestion. The use of multiple separate image files and other associated data, for example often require a client to make multiple requests to the same server in a short amount of time. Therefore an MHP terminal implementation shall include support for persistent connections.

The following section explains how persistent connection is to be used within GEM. The original form of HTTP 1.0 persistent connections is defined as informational text in RFC 2616 [45]. The normative specification text included here is intended to be fully compatible with this RFC.

6.3.7.2.1 HTTP 1.0 persistent connections

When connecting to a server, a GEM terminal shall send the Keep-Alive connection-token:

```
Connection: Keep-Alive
```

An HTTP server (1.0 or 1.1) would then respond with the Keep-Alive connection token and the client may proceed with an HTTP 1.0 (Keep-Alive) persistent connection.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 63 of 861

6.3.7.2.2 The Keep-Alive Header

When the Keep-Alive connection-token has been transmitted with a request or a response, a Keep-Alive header field may also be included. The Keep-Alive header field takes the following form:

```
Keep-Alive-header = "Keep-Alive" ":" 0# keepalive-param
keepalive-param = param-name "=" value
```

where the "0#" notation means that the "keepalive-param" field may be repeated 0 or more times and they are separated by a comma character "," if the field is included more than once.

The Keep-Alive header itself is optional, and is used only if a parameter is being sent. The present document does not define any parameters.

If the Keep-Alive header is sent, the corresponding connection token must be transmitted. The Keep-Alive header must be ignored if received without the connection token.

6.3.7.2.3 GEM and proxies

A GEM terminal must not send the Keep-Alive connection token to a HTTP 1.0 proxy server, as HTTP 1.0 proxy servers do not obey the rules of HTTP 1.1 for parsing the Connection header field.

If a GEM terminal sends Keep-Alive to a proxy server that doesn't understand Connection, which would then erroneously forward it to the next inbound server, which would establish the Keep-Alive connection and result in a hung HTTP 1.0 proxy waiting for the close on the response. The result is that the GEM terminal must be prevented from using Keep-Alive when talking to proxies that doesn't understand Connection.

6.3.7.2.4 Version compatibility

An HTTP 1.1 server may also establish persistent connections with a GEM terminal upon receipt of a Keep-Alive connection token. However, a persistent connection with a GEM terminal can't make use of the chunked transfer coding, and therefore must use a content-length for marking the ending boundary of each message.

For servers used for GEM applications, it is recommended that if HTTP 1.1 servers are used, they should support the HTTP 1.0 persistent connections when initiated by an HTTP 1.0 client using the Keep-Alive connection token.

6.3.7.3 HTTPS

This is described in RFC 2818 [73].

6.3.8 User Datagram Protocol (UDP)

User Datagram Protocol as defined in RFC 768 [47].

6.3.9 DNS

GEM terminals shall implement at least the DNS resolver protocols that enable forward translation of fully qualified domain names to IP addresses as defined by RFC 1034 [64] and RFC 1035 [62] and clarified by RFC 1982 [65] and RFC 2181 [66].

In connection based return channels, where DNS server addresses are provided both from the network as part of connection setup and from a GEM application, those provided from the network shall take precedence.

6.3.10 Additional Transport Protocols

Additional transport protocols that are proprietary and/or specific to a network service provider may be supported through the registration of interaction channel service providers (see org.dvb.spi.ict.InteractionChannelTransportProvider).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

6.4 Transport protocols for application loading over the interaction channel

Three scenarios are envisaged:

- The file system is completely implemented in the broadcast channel (the classic MHP 1.0 model which is not described further here).
- File system implemented only via the interaction channel.
- Hybrid between broadcast stream and interaction channel.

6.4.1 File system implemented only by the interaction channel

This clause addresses the case where the interaction channel presents the only file system, e.g. in MHP [1], where the protocol ID is 0x0003.

GEM terminal specifications may define other protocol ID values that use this mechanism.

6.4.1.1 File system logical structure

The list of items signalled in the transport protocol descriptor(s) (see clause 15.6.1.2.1, "Transport protocol descriptor") are considered to form a single name space.

The GEM terminal when trying to locate a file specified by an incomplete (relative) filename, shall attempt to fetch a file from each of the items in that list in the order in which they are found in the list until the file is found or the list is exhausted.

The items in the list shall either be references to .zip files (see ZIP [67]) or base URLs ending in "slash" (/) onto which the path of the requested file shall be concatenated. Any items in the list which are not one of these two types shall be ignored. Errors in discovering whether a specific file can be reached through a specific list item shall be ignored. The list must contain at least one item.

As an example, consider the GEM platform retrieving the "dvb.fontindex" file for an application from an interactive channel for which the AIT File includes two transport protocol descriptors with the following contents:

Table 1: Example transport descriptor selector byte payload

URL base	URL extension
"http://www.dvb.org"	"applications/application1.zip"
	"graphics/"
	"shared/utils.zip"
"http://www.ebu.ch"	"general/misc.zip"
"http://www.dvb.org"	"other_stuff/we_dont_use_this_very_often.zip"

Then the search order for "dvb.fontindex" would be the following:

a) In the contents of http://www.dvb.org/applications/application1.zip.

b) From the URL http://www.dvb.org/graphics/dvb.fontindex.

c) In the contents of http://www.dvb.org/shared/utils.zip.

d) In the contents of http://www.ebu.ch/general/misc.zip.

e) In the contents of http://www.dvb.org/we_dont_use_this_very_often.zip.

The search order for "abc/dvb.hashfile" would be the following:

a) For "abc/dvb.hashfile" in the contents of http://www.dvb.org/applications/application1.zip.

b) From the URL http://www.dvb.org/graphics/abc/dvb.hashfile.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

c) For "abc/dvb.hashfile" in the contents of http://www.dvb.org/shared/utils.zip.

d) For "abc/dvb.hashfile" in the contents of http://www.ebu.ch/general/misc.zip.

These file system semantics shall be followed for all files fetched by or on behalf of a GEM application. This includes the files whose names and locations are standardized by the present document, e.g. 'dvb.fontindex', 'dvb.hashfile', 'dvb.signature.x' and dvb.certificate.x'.

For DVB-J, with the following application location descriptor:

a) base_directory "/app1/starting".

b) classpath_extension "/shared;/app1/later".

c) initial_class "org.dvb.demo.myXlet".

The search order for a class would be the following:

a) For "app1/starting/org/dvb/demo/myXlet.class" in the contents of http://www.dvb.org/applications/application1.zip.

b) From the URL http://www.dvb.org/graphics/app1/starting/org/dvb/demo/myXlet.class.

c) For "app1/starting/org/dvb/demo/myXlet.class" in the contents of http://www.dvb.org/shared/utils.zip.

d) For "app1/starting/org/dvb/demo/myXlet.class" in the contents of http://www.ebu.ch/general/misc.zip.

e) other_stuff/we_dont_use_this_very_often.zip.

f) For "/app1/starting/shared/org/dvb/demo/myXlet.class" in the contents of http://www.dvb.org/applications/application1.zip.

g) From the URL http://www.dvb.org/graphics/app1/starting/shared/org/dvb/demo/myXlet.class.

h) For "/app1/starting/shared/org/dvb/demo/myXlet.class" in the contents of http://www.dvb.org/shared/utils.zip.

i) For "/app1/starting/shared/org/dvb/demo/myXlet.class" in the contents of http://www.ebu.ch/general/misc.zip.

j) other_stuff/we_dont_use_this_very_often.zip.

k) For "/app1/starting/app1/later/org/dvb/demo/myXlet.class" in the contents of http://www.dvb.org/applications/application1.zip.

l) From the URL http://www.dvb.org/graphics/app1/starting/app1/later/org/dvb/demo/myXlet.class.

m) For "/app1/starting/app1/later/org/dvb/demo/myXlet.class" in the contents of http://www.dvb.org/shared/utils.zip.

n) For "/app1/starting/app1/later/org/dvb/demo/myXlet.class" in the contents of http://www.ebu.ch/general/misc.zip.

o) other_stuff/we_dont_use_this_very_often.zip.

6.4.1.2 File transfer

Files to be transferred from an "http" URL shall be transferred using either HTTP 1.1 as defined in clause 6.3.7.1, "HTTP 1.1" or the profile of HTTP specified under clause 6.3.7.2, "GEM profile of HTTP 1.0".

Files to be transferred from an "https" URL shall be transferred as defined in clause 6.3.7.3, "HTTPS".

For other URL schemes, if a registered interaction channel transport protocol service provider exists for the relevant URI scheme, that provider shall be used.

NOTE: That transport protocol service providers may not be registered for the http: or https: URI schemes (i.e. service providers cannot override the GEM implementation's internal implementation of the HTTP and HTTPS protocols).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

6.4.1.3 Class encoding

The classes of DVB-J applications can be delivered either as discrete class files or within ZIP files (see ZIP [67]).

NOTE 1: This contrasts to the broadcast case where object carousel files are simply class files.

The classes of DVB-J applications shall be delivered as discrete class files at the level of the file system.

NOTE 2: This means that DVB-J class files are allowed to be held on an HTTP server either as discrete class files or in ZIP files (see ZIP [67]) where ZIP files form part of the overall file system.

6.4.1.4 Directory listing in this file system

For the authentication (see clause 12.4.1.5, "Special authentication rules"), this file system shall be considered to be one not supporting directory listing.

For listing the files in a directory (see clause 14.7, "Files and file names"), this file system shall be considered to be one where the MHP terminal is never certain of the complete contents of a directory.

6.4.2 Hybrid between broadcast stream and interaction channel

In the hybrid case all directory information is provided in the broadcast stream but some, or possibly all, of the file contents are provided via the interaction channel.

6.4.2.1 File transfer

6.4.2.1.1 Broadcast file delivery

In GEM terminals that use the DSM-CC User-to-User Object Carousel, the file contents are carried via a `BIOP::File` as is the normal case for a DSM-CC User-to-User Object Carousel as described in clause 6.2.5, "Object carousel". The IOR from the file binding to the file contents uses either a `BIOPProfileBody` or a `LiteOptionsProfileBody`.

6.4.2.1.2 Interaction channel delivery

In GEM terminals that use the DSM-CC User-to-User Object Carousel, the IOR from the file binding to the file contents uses the `HTTPProfileBody` (see table 2) to identify the location of the file contents on the interaction channel.

This form of IOR shall only be used for `BIOP::File` objects.

See clause 6.3.7.2, "GEM profile of HTTP 1.0" for the profile of HTTP that is used to retrieve the contents of the file.

6.4.2.1.3 HTTPProfileBody

The HTTP Profile body specifies `host`, `port` and `path_segments`. In HTTP requests these are concatenated to form a URL of the form:

```
http://host:port/path_segments
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 67 of 861

Table 2: HTTP Profile Body syntax

Syntax	bits	Type	Value	Comment
HTTPProfileBody {				
profileId_tag	32	uimsbf	0x44564200	
profile_data_length	32	uimsbf	*	
profile_data_byte_order	8	uimsbf	0x00	big endian byte order
version.major	8	uimsbf	0x01	protocol major version 1
version.minor	8	uimsbf	0x00	protocol minor version 0
host_data_length	8	uimsbf	N1	
for (k=0; k<N1; k++) {				
host_data	8	uimsbf	+	
}				
port	16	uimsbf		
objectKey_length	16	uimsbf	N2	
for (k=0; k<N2; k++) {				
objectKey_data	8	uimsbf	+	
}				
}				
NOTE: The tag values 0x44564201 to 0x44564201 have been reserved for future use by the DVB. 0x44 is ASCII for "D", 0x56 is "V" and 0x42 is "B".				

version: These 8-bit fields indicate the version of the protocol that the server will use to deliver the file specified. The version value 1.0 indicates that the transport protocol is as defined in clause 6.3.7.2, "GEM profile of HTTP 1.0".

host_data: These bytes convey the identifier of the Internet host to which messages may be sent. It may be a fully qualified domain name or Internet standard "dotted decimal" form (e.g. "192.231.79.52").

port: This 16 bit unsigned integer is the TCP/IP port number at the specified host where the target agent is listening for requests.

objectKey_data: These bytes form a string which carries the path_segments portion of the URL which uniquely identifies the object on the server RFC 2396 [46].

6.5 IPTV protocols

GEM terminal specifications supporting the IPTV target shall define protocols for the delivery of GEM services and service information. GEM terminal specifications may mandate support for any of the protocols specified by the following subclauses.

NOTE: GEM-IPTV protocols may rely on the provider interfaces, defined in clause 9.11, "Providers".

6.5.1 Transport protocols

6.5.1.1 Service Discovery and Selection

The protocols for unicast and multicast delivery of service discovery and selection information are defined in clause 5.4 of TS 102 034 [80].

6.5.1.2 Broadband Content Guide

The protocols for unicast and multicast delivery of broadband content guide information are defined in clause 4 of TS 102 539 [81].

6.5.1.3 Real Time Protocol (RTP)

The use of RTP is defined in clause 7 of TS 102 034 [80].

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

6.5.1.4 Real Time Streaming Protocol (RTSP)

The use of RTSP is defined in clause 6 of TS 102 034 [80].

6.5.1.5 Internet Group Management Protocol (IGMP)

The use of IGMP is defined in clause 7 of TS 102 034 [80].

6.5.2 Service information and metadata protocols

6.5.2.1 IP service discovery

This is defined by clause 5.2 of TS 102 034 [80] excluding clause 5.2.6.6.

GEM terminals shall monitor and detect changes in this signalling however there is no requirement to do this more frequently than once per day.

6.5.2.2 Broadband content guide

This is defined by clauses 5, 6 and 7 of TS 102 539 [81] and clause 5.2.6.6 of TS 102 034 [80].

6.6 OTT Protocols

6.6.1 Protocols for streaming

Unicast streaming using HTTP 1.1 [45] shall be supported for the OTT target.

NOTE: HTTP 1.1 support the range header in a GET request, which allows partial retrieval for use in cases such as trick play or seek operations to reduce unnecessary network usage.

To optimize the streaming user experience over best-effort broadband lines, the receiver shall implement proper buffering and playback strategies to cope with varying network conditions. The details of such strategies are implementation dependant. Adaptive streaming is one possibility to address bandwidth variations.

6.6.1.1 Adaptive Streaming

The user experience with Streamed CoD based on a single instance (single coding rate) of a given content may be adversely affected by the different and varying conditions of each customer's unmanaged broadband access. Different because of possible different access line profiles (either commercial or technical); varying because of the continuously changing end-to-end network conditions.

Several solutions have been recently announced or introduced into the market (e.g. Adobe, Microsoft, Apple, Widevine, 3GPP, MPEG, OIPF), but there is not yet a standardized and DVB-endorsed solution for adaptive streaming.

A common mechanism for adaptive streaming is when a service provides a content item in multiple bitrates in a way that enables a terminal to adapt to variations in the available bandwidth by seamlessly switching from one version to another, at a higher or lower bitrate, while receiving and playing the content.

This is achieved by encoding a content item in alternative representations of different bitrates and segmenting these representations into temporally aligned and independently encoded segments.

Typically adaptive streaming solutions provide a streaming manifest file describing the available multimedia content segments at various bit-rates. This information is used within the JMF player to select the appropriate segments for the current network condition.

To optimize the user experience over broadband lines, the receiver must implement buffering and playback to cope with varying network conditions. Buffering and adaptation to changing network conditions should be handled at platform and protocol level, but applications are able to provide customers with some rough technical information about content and/or network and/or client characteristics/status in the context of Streamed CoD (see Monitoring API in Annex N.2).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Since GEM is protocol-agnostic, no assumptions are made on the particular manifest file supported by a given platform. GEM does not mandate a specific adaptive streaming protocol.

6.6.2 Protocols for download

If content download is supported, HTTP 1.1 [45] shall be supported. The content download mechanism is agnostic to the file formats that are transferred.

NOTE: HTTP 1.1 supports the range header in a GET request, which allows partial retrieval for continuing failed or interrupted download requests.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

7 Content formats

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

7.1 Static formats

7.1.1 Bitmap image formats

7.1.1.1 Image encoding restrictions

Any indications in the transmitted image with respect to pixel scaling, colour space or gamma are to be ignored in the presenting of the image. One image pixel shall be mapped to one graphics pixel in the current graphics configuration, unless otherwise scaled by the application directly.

See also clause 7.5, "Colour representation".

7.1.1.2 JPEG

JPEG as defined in ISO/IEC 10918-1 [34] using the JFIF [42] file exchange format.

Only coding using sequential DCT-based mode or progressive DCT-based mode is required to be supported by implementations.

Specifically, lossless and hierarchical modes and arithmetic coding of DCT coefficients need not be supported.

The thumbnail feature of JFIF [42] need not be supported. GEM terminals not supporting thumbnails shall skip them if present and continue decoding the rest of the image.

See also clause 15.3, "JPEG - restrictions" for other possible restrictions.

7.1.1.3 PNG

PNG is defined as in PNG Version 1.0 [43] with the following notes and modifications:

- Later versions of PNG introduce additional chunk types, which are not required to be supported
- See also clause 15.1, "PNG - restrictions".
- GEM terminal specifications may allow or require processing of colour space or gamma information in image transformations.

7.1.1.4 GIF

GIF is defined as in GIF 89a [33].

7.1.2 MPEG-2 I-Frames

MPEG-2 I-Frames are defined as in ISO/IEC 13818-2 [37].

The payload of a file delivering an MPEG-2 I frame shall:

- Be a valid video_sequence() including a sequence_extension().
- Contain one I frame only, i.e. one picture_header(), one picture_coding_extension(), and one picture_data() encoded as an intra coded frame, with picture structure = "frame".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

That is the structure is:

```
sequence_header()
sequence_extension()
extension_and_user_data(0)
optional group_of_pictures_header() and extension_and_user_data(1)
picture_header ( picture_coding_type = "I frame")
picture_coding_extension ( picture_structure = "frame picture")
extension_and_user_data(2)
picture_data()
sequence_end_code()
```

MPEG I-frames shall be presented with no DecFC (except for any scaling) i.e. the raster's encoded aspect ratio and any AFD shall be ignored.

7.1.3 MPEG-2 Video "drips"

The drip feed mode consists of letting an application progressively feed the MPEG-2 video decoder with chunks of an MPEG-2 video stream. In this mode, it is only required for the decoder to handle I and P frames (i.e. not B frame). Each chunk shall contain one frame and a certain number of syntactic elements (as described in ISO/IEC 13818-2 [37]) such as `sequence_header()` or `group_of_picture_header()`.

Firstly, the content of each of the chunks of bytes fed to the decoder shall comply with the following syntax:

```
optional {
        sequence_header()
        sequence_extension()
        extension_and_user_data(0)
        optional {
                group_of_pictures_header()
                extension_and_user_data(1)
        }
}

picture_header ( picture_coding_type = "I frame" or "P frame")
picture_coding_extension ( picture_structure = "frame picture")
extension_and_user_data(2)
picture_data()
optional {
        sequence_end_code()
}
```

In addition, the overall concatenation of chunks over time shall respect the authorized combinations of syntactic elements described in ISO/IEC 13818-2 [37] to build a legal MPEG-2 video stream. The following diagram, extracted from ISO/IEC 13818-2 [37], reflect the rules defined in that standard:

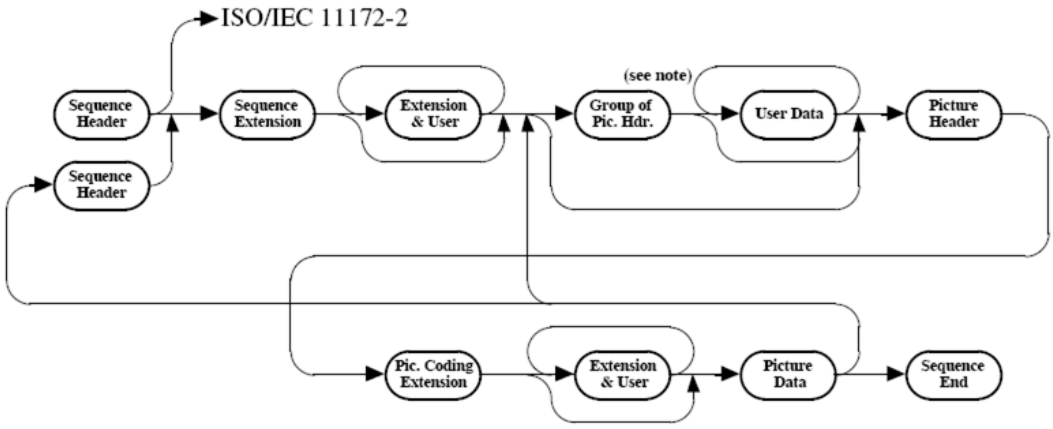


Figure 12: MPEG-2 Video Stream

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

NOTE 1: After a GOP the first picture is an I-picture.

The following restrictions are applied to P-frames:

- The P-frame shall contain no prediction information (i.e. no motion vector shall be present in macroblock elements).
- The allowed macroblock_types for P-frames are:
  - "Intra" (i.e. VLC code 0001 1).
  - "Intra, Quant" (i.e. VLC code 0000 01).
  - "No MC, Coded" (i.e. VLC code 01).
  - "No MC, Coded, Quant" (i.e. VLC code 0000 1).

NOTE 2: The standard semantics for P-frames allow `macroblock_escape` and `macroblock_address_increment` to signal skipped macroblocks. This allows P-frames to be very sparse, only carrying macroblocks positioned at certain locations on the screen. This contrasts with semantics for an I-frame where macroblocks are required to fill the full screen.

If invalid content is fed to the MPEG-2 video decoder, the content is discarded and there are no guarantees when subsequent valid chunk of byte fed to the decoder will be displayed (unless the decoder is restarted).

This mode requires the decoder to be in the "low delay" mode as defined in ISO/IEC 13818-2 [37].

This mode can be used by connecting a `org.dvb.media.DripFeedDataSource` instance to a Player representing a MPEG-2 video decoder. See annex N, "Streamed media API extensions".

MPEG video drips shall be presented with no DecFC (except for any scaling) i.e. the raster's encoded aspect ratio and any AFD shall be ignored.

7.1.4 Monomedia format for audio clips

The format for audio clips is MPEG-1 Audio (Layer 1 and 2) ES data as defined as in ISO/IEC 11172-3 [35] and constrained in TS 101 154 [2].

Each "file" of audio content is a binary data file carrying Audio elementary stream data. Each "file" delivers an integer number of audio access units and the first byte of each file is the first byte of an audio access unit. The MPEG Audio data in all other respects conforms to the specifications provided in TS 101 154 [2].

Implementations decoding audio clips can assume that they have an approximately constant number of bytes per second. If this not true then the behaviour is implementation dependent.

Use of the MPEG-1 format as defined above is not required for GEM terminal specifications for packaged media targets. This clause shall apply to GEM terminal specifications that include the functional equivalent named "Audio Clips" as specified in clause 15.6, "Functional equivalents". GEM terminal specifications that do not include this definition shall specify a functional equivalent that satisfies the requirements of the APIs mentioned in the table as included in clause 15.2, "Minimum media formats supported by DVB-J APIs".

7.1.5 Monomedia format for text

Java modified UTF-8 as defined in the specification of the method `java.io.DataOutput.writeUTF`.

NOTE: Based on ISO/IEC 10646-1 [15] but modified with respect to the encoding of the character code zero.

7.1.5.1 Built-in character set

See annex E, "Character set".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

7.2 Media streaming formats

7.2.1 Audio

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

7.2.2 Video

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

7.2.3 Subtitles

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

NOTE: OCAP [5] does not include support for subtitles. It does include US closed-captioning which is somewhat similar, but has different regulatory requirements and usage models.

The content formats supported for subtitles are:

- DVB Subtitles.
- Teletext.

See clause 13.5, "Subtitles".

In the event that both DVB Subtitles and DVB Teletext are available then DVB Subtitles will take precedence (i.e. if a stream is flagged as having both DVB Subtitles and Teletext Subtitles then the DVB Subtitles will be displayed).

Teletext Subtitles conform to the same display model as DVB subtitles.

Application control and detection of subtitles, whether they be DVB Subtitles or Teletext Subtitles, will be through JMF. The application will have no knowledge of the delivery/presentation protocol being used to provide subtitles.

No APIs will be provided to access Teletext data packets and no timing model is provided for the decoding of Teletext subtitle. Text subtitles will be decoded as soon as the data becomes available.

7.2.3.1 DVB Subtitles

DVB Subtitles are defined as in EN 300 743 [i.7].

7.2.3.2 Teletext

Transmission of the text is as defined in EN 300 472 [i.6]. The data format is as defined in ETS 300 706 [56] but restricted to presentation level 1,5 or lower. Signalling of the Teletext subtitle page will be via the Teletext descriptor as defined in EN 300 468 [14].

Within the GEM specification Teletext is only supported as an alternative content format for delivery of subtitles. The GEM specification does not address its possible use as a navigable content format.

NOTE: Manufactures remain free to implement full Teletext support based on regulatory requirement or market demand. Such support would be implemented outside of the GEM environment, by VBI re-insertion of the non-subtitle text or through a native Teletext Decoder. The user interface integration is then an issue for the manufacturer to resolve.

It is envisaged that broadcasters will use GEM applications to deliver navigable text services providing a greater level of interactivity and enhanced graphics.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

7.2.4 Containers

A Container is what in MPEG is called the “system layer”, i.e. a means for multiplexing and synchronizing multiple media streams with associated system information (metadata).

Examples of widely used containers are MPEG-2 TS on the broadcast side and MP4 on the internet for streaming purposes.

At least one container format shall be specified in a GEM terminal specification.

7.2.5 Streaming Manifest

A “streaming manifest” is a file, which can be used to tell the receiver that several alternative representations are available on the server for the same content, e.g. more files corresponding to encodings at different bit rates of the same content which the receiver can seamlessly switch to based on varying network conditions.

A GEM terminal, which supports adaptive streaming, SHALL select or define at least one file format for the straming manifest. URLs which reference manifest files whose formats are supported by the implementation SHALL be accepted when creating a JMF player.

If a streaming manifest file is referenced in the URL used by an application to create a JMF player which does support that particular manifest format, the JMF player will perform all the relevant operations (manifest parsing, switching from one representation to another based on certain conditions, …) without any application involvement.

GEM does not put any restrictions on the format of the streaming manifest.

7.3 Resident fonts

The inclusion of resident fonts is not required for GEM terminal specifications for packaged media targets.

This clause shall apply to GEM terminal specifications that include the functional equivalent named "Resident Fonts" as specified in clause 15.6, "Functional equivalents".

For GEM terminal specifications that do not include the resident font there shall be signalling to indicate a font file packaged with an application that will be available under the logical name "SansSerif", e.g. returned by `java.awt.Toolkit.getFontList()` or provided to the `java.awt.Font` constructor.

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

See clause G.4, "Resident fonts and text rendering".

See also annex D, "Text presentation".

7.4 Downloadable fonts

A GEM terminal specification may define a way to check the name, style and size parameters of a `FontFactory.createFont()` call on validity with respect to the data present in the font file from which the font should be created.

A GEM terminal specification may require this check for a `FontFactory` constructed for a single font file.

A GEM terminal specification may require this check for `FontFactory` constructed for a font index file in addition to the check for support of the given name, style and size in the given font index file.

If such a check is not defined for a GEM terminal specification then the parameters used in a `createFont()` call shall not be checked against data present in the font file.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 75 of 861

7.4.1 PFR

Use of the PFR downloadable font format as defined here is not required for GEM terminal specifications for packaged media targets.

This clause shall apply to GEM terminal specifications that include the functional equivalent named "Downloadable Fonts" as specified in clause 15.6, "Functional equivalents". GEM terminal specifications that do not include this definition shall specify a functional equivalent that enables a font to be packaged with GEM applications. This clause is an example of such a format, and is informative for these GEM terminal specifications.

PFR0 (Portable Font Resource version 0) is defined as in DAVIC 1.4.1p9 [17] as the coding format for fonts. Receivers implementing a GEM specification that requires PFR support are only required to provide support for the outline version of the font.

The `charCode` value in the PFR `charRecord` and `bmapCharRecord`, and the `charCode1` and `charCode2` fields of the pairKernData() structure, shall be the ISO/IEC 10646-1 [15] code for the glyph. If these fields are 16-bit, they are encoded using UCS-2 and shall not be a high-surrogate or a low-surrogate code value. If these fields are 8-bit, then the one-byte character code shall be the least significant 8 bits of a 16-bit UCS-2 code, where the most significant 8 bits are all zero.

For fonts in the PFR format, the font name shall be the `fontID` field in the font file. The `fontID` field in the font file shall be encoded using the Java modified UTF-8 encoding (see clause 7.1.5, "Monomedia format for text").

Each font in a PFR file can have auxiliary data in its physical font record. If present this auxiliary data shall adhere to the syntax as specified in table 3. It shall consist of a number of blocks terminated by a block of length 0 and type 0.

Table 3: PFR auxiliary data

	No.of Bits	Identifier
auxDataFontRecord() {		
do {		
blockLength	16	uimsbf
blockType	16	uimsbf
switch (blockType) {		
case 2:		
for (i = 0; i < 10; i++) {		
reserved	8	uimsbf
}		
ascent	16	tcimsbf
descent	16	tcimsbf
reserved	32	uimsbf
externalLeading	16	tcimsbf
for (i = 0; i < (blockLength - 24); i++) {		
reserved	8	uimsbf
}		
break;		
default:		
for (i = 0; i < (blockLength - 4); i++) {		
reserved	8	uimsbf
}		
break;		
}		
} while (blockLength > 0 || blockType != 0)		
}		

An auxiliary data font record consists of a number of blocks terminated by a block of length 0 and type 0.

blockLength: The total number of bytes in this block, including the blockLength and the blockType.

blockType: A number that defines the type of data in the block. 0x0000 to 0x7fff are reserved, 0x8000 to 0xffff are user defined types.

ascent: Represents the font ascent, which is the distance from the base line to the top of most alphanumeric characters.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

descent: Specifies the descent (units below the base line) of most alphanumeric characters.

externalLeading: Specifies the amount of extra leading (space) that the application adds between rows. The designer may set this member to zero.

GEM terminals shall ignore the contents of the reserved fields within block type 2 for the purposes of computing font metrics. GEM terminal behaviour for block types other than 2 is implementation dependent and these should not be used by GEM applications.

7.4.2 OpenType

OpenType is standardized in ISO/IEC 14496-18 [75] as the font format for MPEG-4, and is a full-featured font format that supports advanced typographic features, multi-lingual text processing and international character sets.

ISO/IEC 14496-18 [75] defines different MPEG text profiles and levels of functionality. GEM receivers shall support OpenType fonts with TrueType outlines, as defined by Advanced Simple Text Profile at Level 2 in ISO/IEC 14496-18 [75], section 6.2.5. OpenType fonts with TrueType outlines enable the highest quality of text rendering on low-resolution displays.

For OpenType fonts, the font name used by the GEM application shall be the full font name specified in the Naming ('name') table of OpenType font with the `NameID = 4`.

See also clause D.2.2, "Downloaded fonts".

7.5 Colour representation

7.5.1 Background (informative)

The method of colour encoding is critical to how consistently the colours in an image can be reproduced across different systems. The description must be cast in a way which is independent of the mechanisms by which it will finally be reproduced for the viewer.

The International Colour Consortium (ICC) has proposed a thorough solution to the precise communication of colour in open systems. However the ICC profile format is somewhat over-specified for GEM. The ICC mechanism for ensuring that a colour is correctly mapped from an input to the output colour space is by attaching a profile for the input colour space to the image in question. This is appropriate for high end systems, especially those in the print media. However, a primarily CRT based home platform neither needs, nor has the processing power and available bandwidth, to handle an embedded profile mechanism. It would also require some sophistication on the part of the end consumer to set up properly.

Fortunately by adopting a single default colour space that can be processed as an implicit ICC profile the advantages of the ICC approach are gained, and the system is later scalable to a full colour management system with a clear relationship to existing ICC colour management systems while minimizing software and support requirements in a GEM terminal today.

A colour space is a model for representing colour numerically in terms of three or more coordinates or tristimulus values. An RGB colour space represents colours in terms of Red, Green and Blue coordinates. The MHP format shall use the specific RGB encoding for colour imagery, sRGB as defined in IEC 61966-2-1 [39]. This is suitable for a wide range of presentation environments including TV's and has become widely adopted in the computer environment and WWW. It is, for example, compatible with CCIR Recommendation ITU-R Recommendation BT.709-5 [41] standard for colour encoding in HDTV. This format has the advantage of device independence without a great deal of additional overhead.

For sRGB, the goal is to communicate the appearance of colours as displayed on a reference monitor in terms of 8-bit digital code values for each coordinate. sRGB colour values represent colour appearance with respect to a defined reference viewing environment.

For colour stimuli viewed in the reference viewing environment, sRGB values are defined by a series of simple mathematical operations from standard CIE colourimetric values.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The sRGB format is a good match for 24 bit colour on most CRTs. In devices where a great deal of damage is done to the colour space it may not give consistent results. For example dithering to a 4-bit per primary colour map will violate the gamma assumptions.

7.5.2 Specification

All images transmitted shall be within the gamut encompassed by the sRGB colourspace. Where possible this should be coded so that the terminal does not have to translate. Where this is impractical the sRGB image may be transcoded into a different colourspace provided the gamut assumption is not violated (i.e. to be consistent with JFIF, JPEG images shall be sent in the region of the YC_rC_b colourspace that overlaps with the sRGB gamut).

NOTE: That the presentation of images using colours outside of the sRGB gamut is platform dependent.

Images created in GEM will be in the sRGB colourspace by default, although manufacturers are free to provide support for other colour spaces if they choose. All MHPs shall support transformations from sRGB to the colour spaces allowed by the MPEG-2 definition (e.g. ITU-R Recommeandations BT.709-5 [41] and BT.470-6 [40]) and vice versa, manufacturers may choose to support transformations to and from other colour spaces.

7.5.2.1 The sRGB Reference Viewing Environment

The reference display conditions and viewing environment for sRGB are partly described in table 4. A reference viewing environment must be provided to allow for the unambiguous definition of colour, the sRGB reference viewing environment corresponds to conditions typical of indoor viewing of CRTs - further details can be found in the IEC 61966-2-1 [39].

The sRGB reference conditions therefore provides a well defined reference compatible with ITU-R Recommendation BT.709-5 [41].

Table 4: sRGB reference Display conditions

Condition	sRGB
Viewing flare	1,0 %
Reference Background	20 %
Display model Offset	0,055
Display Gun/Phosphor Gamma	2,4
Display white point	x = 0,3127 y = 0,3290 (D65 Hunt, R.W.G. [27])
Ambient Lighting	64 lx
Display Luminance level	80 cd/m^2

7.5.2.2 Colourimetric Definitions and Encodings

sRGB tristimulus values can be computed as follows, firstly linear sRGB tristimulus are computed as linear combinations of the 1931 CIE XYZ (CIE 15 [24]) values using the following relationship.

$$\begin{bmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{bmatrix} = \begin{bmatrix} 3,2410 & -1,5374 & -0,4986 \\ -0,9682 & 1,8760 & -0,0416 \\ 0,0556 & -0,2040 & -1,0570 \end{bmatrix} \begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix}$$

In the encoding process, negative sRGB tristimulus values, and sRGB tristimulus values greater than 1,00 are not retained. The luminance dynamic range and colour gamut of sRGB is limited to the tristimulus values between 0,0 and 1,0 by simple clipping. This gamut, however, is large enough to encompass most colours that can be displayed on CRT monitors.

For comparison, the CIE chromaticities for the red, green, and blue ITU-R Recommendation BT.709-5 [41] and ITU-R Recommendation BT.470-6 [40] reference primaries, and for CIE Standard Illuminant D65 (IEC 61966-2-1 [39]), are given in tables 5 and 6. From these primaries the YC_bC_r transmitted values are computed by similar relationships.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Therefore ITU-R Recommendation BT.709-5 [41] YC_bC_r colourspace and similar video colour spaces can be converted to sRGB and vice versa by way of CIE XYZ (CIE 15 [24]). Chromaticities for other video formats allowed in MPEG streams can be found in their respective standards.

Table 5: ITU-R Recommendation BT.709-5 [41] reference primaries and CIE standard illuminant

	Red	Green	Blue	D65
x	0,6400	0,3000	0,1500	0,3127
y	0,3300	0,6000	0,0600	0,3290
z	0,0300	0,1000	0,7900	0,3583

Table 6: ITU-R Recommendation BT.470-6 [40] reference primaries and CIE standard illuminant

	Red	Green	Blue	D65
x	0,6700	0,2100	0,1400	0,3100
y	0,3300	0,7100	0,0800	0,3160
z	0,0100	0,0800	0,7800	0,3740

The linear sRGB tristimulus values are next transformed to non linear sR'G'B' values. This process closely approximates the effect of a "gamma" curve of 2,2 with a slight offset. This makes sRGB consistent with legacy systems and images.

if $R_{sRGB} > 0{,}00304$ and $G_{sRGB} > 0{,}00304$ and $B_{sRGB} > 0{,}00304$

then

$R'_{sRGB} = 1{,}055 \times R_{sRGB} \text{ ^ } (1{,}0/2{,}4) - 0{,}055$

$G'_{sRGB} = 1{,}055 \times G_{sRGB} \text{ ^ } (1{,}0/2{,}4) - 0{,}055$

$B'_{sRGB} = 1{,}055 \times B_{sRGB} \text{ ^ } (1{,}0/2{,}4) - 0{,}055$

else

$R'_{sRGB} = 12{,}92 \times R_{sRGB}$

$G'_{sRGB} = 12{,}92 \times G_{sRGB}$

$B'_{sRGB} = 12{,}92 \times B_{sRGB}$

end if

Finally, the non-linear sR'G'B' values are converted to digital code values. This conversion scales the sR'G'B' values by using the equation below where WDC represents the white digital count and KDC represents the black digital count.

- $R_{8bit} = ((WDC - KDC) \times R'_{sRGB}) + KDC.$
- $G_{8bit} = ((WDC - KDC) \times G'_{sRGB}) + KDC.$
- $B_{8bit} = ((WDC - KDC) \times B'_{sRGB}) + KDC.$

The current sRGB specification uses a black digital count of 0 and a white digital count of 255 for 24-bit (8-bits/channel) encoding, and the MHP shall adopt the same convention.

NOTE: Some digital video signals may use a black digital count of 16 and a white digital count of 235 in order to provide a larger encoded colour gamut.

Details of the reverse transformation from sRGB to CIE XYZ are given in IEC 61966-2-1 [39], mappings from ITU-R Recommendation BT.709-5 [41] and ITU-R Recommendation BT.470-6 [40] to CIE XYZ are given in ISO/IEC 13818-2 [37].

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 79 of 861

7.6 MIME types

Table 7: File type identification

MIME type	Extension (note)	Definition of content
"image/jpeg"	".jpg"	As defined in clause 7.1.1.2, "JPEG".
"image/png"	".png"	As defined in clause 7.1.1.3, "PNG" possibly with a constrained profile as defined in clause 15.1, "PNG - restrictions".
"image/gif"	".gif"	As defined in clause 7.1.1.4, "GIF".
"image/mpeg"	".mpg"	As defined in clause 7.1.2, "MPEG-2 I-Frames".
"video/mpeg"	".mpg"	As defined in clause 7.2.2, "Video".
"video/vnd.dvb.mpeg.drip"	".drip"	As defined in clause 7.1.3, "MPEG-2 Video "drips"".
"audio/mpeg"	".mp2"	As defined in clause 7.1.4, "Monomedia format for audio clips" or as defined in clause 7.2.1, "Audio".
"text/vnd.dvb.utf8"	".txt"	As defined in clause 7.1.5, "Monomedia format for text".
"text/xml"	".xml"	Not defined in the present document.
"text/css"	".css"	Not defined in the present document.
"text/plain"	".txt"	As defined in clause 7.1.5, "Monomedia format for text".
"text/ecmascript"	".js"	Not defined in the present document.
"image/vnd.dvb.subtitle"	".sub"	As defined in clause 7.2.3, "Subtitles".
"text/vnd.dvb.subtitle"		
"text/vnd.dvb.teletext"	".tlx"	As defined in clause 7.2.3.2, "Teletext".
"application/vnd.dvb.pfr"	".pfr"	As defined in clause 7.4, "Downloadable fonts".
"application/vnd.dvbj"	".class"	A DVB-J class file.
“application/vnd.dvb.ait”	“.ait”	AIT File as defined in clause 10.4.5, “AIT File”
“application/vnd.dvb.ait+xml”	“.aitx”	XML encoding of the AIT as defined in Annex AR, “XML encoding for AIT”
"multipart/vnd.dvb.service"	".svc"	An MPEG Program (DVB Service) conforming to DVB norms.
NOTE: Future formats may use more characters in the extension.		

GEM terminal specifications may replace the entry "application/vnd.dvb.pfr" with "application/font-tdpfr" as defined in RFC 3073 [12] if they wish.

Earlier versions of GEM and derived specifications defined the DVB MIME types without the vnd. prefix. These type names are deprecated and their use is strongly discouraged. If a GEM terminal gets a deprecated MIME type, it shall behave, as if the vnd. prefix was present.

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

7.6.1 Rationale

The MIME types are defined to reserve a name space for the possible future support of downloadable JMF players.

The file name extensions shall be included in broadcasts to assist receivers identify the type of the content. For DVB-J applications, this is described in table 30, "Return types of URL.getContent()".

Not all MIME and filename extensions defined in the above table are actually used in the present document. For DVB-J, the APIs which consume media types are described in clause 15.2, "Minimum media formats supported by DVB-J APIs". With MIME types whose corresponding media is not listed for a particular profile, access to that content type from files is not defined for that profile.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

8 Void

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

9 Application model

9.1 Service-bound GEM applications

9.1.1 Basic lifecycle control

The basic control of the lifecycle of service-bound GEM applications is through the selection of services. Selection of a service can be initiated by the user of the GEM terminal using the navigator as well as by GEM applications offering service selection functionality. For broadcast and IPTV applications, service selection functionality usually takes the form of an EPG; EPG functionality is not typical for packaged media.

The unit for the presentation and execution of content in the GEM specification is the service. A service in GEM represents a group of pieces of content which are intended to be presented together to the end-user. A service may consist of the contents of a service, including audio/video streams, data streams and all the service information, applications and application signalling that is being broadcast. For packaged media targets, a service is made up of elements that are typically stored on a storage medium, including audio/video streams, data streams, service information, applications and application signalling. Other forms of service are possible, such as a stored service (see clause 9.6.2, "Stored services").

Every service that gets presented by a GEM platform is presented within a service context. These form one of the foundations for the runtime environment and the execution model. A service context is an "environment" in which a service gets presented. It defines the boundaries of the service (letting the platform and applications identify which of the pieces of content that are being presented make up a given service). It also enables that service to be addressed and controlled as a single entity.

In a DVB-J application, a service context is represented by an instance of the `ServiceContext` class. Where multiple DVB-J applications are being presented in the same service context, the number of `ServiceContext` objects representing a service context is implementation dependent, but each application sees only one such instance. Changes made by one application to the `ServiceContext` object that it has are visible to the `ServiceContext` objects representing the same service context in other applications. DVB-J applications may obtain a reference to the service context within which they are executing through using the method `getServiceContext(XletContext)` on the `ServiceContextFactory` class.

The means by which the navigator of a GEM terminal supports selection of services is implementation dependent. However, where a GEM application is using the numeric keys of the remote control, the navigator shall not respond to the user pressing these keys by causing service selection. Hence the user pressing the numeric keys to enter his pincode does not cause service selection.

A service context can present only one service at any one time. Selecting a service to be presented in a service context causes any previous service being presented in that service context to stop being presented. Any content part of the previous service which is not part of the new service shall stop being presented. GEM terminals may limit the number of broadcast service contexts which can be presented simultaneously.

- In a DVB-J application, selecting a service corresponds to calling the `select()` method on an instance of `javax.tv.service.selection.ServiceContext`.
- Host Control tune requests from a CI module cause service selections. Host Control replace / clear_replace has an equivalent effect to using `javax.tv.media.MediaSelectControl`.

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

9.1.2 Starting applications

When a service is selected, applications which are listed in the Application Description of the service and identified as auto-start shall be launched as described in clause 10.4.3.1 without explicit intervention of the user. Applications which are started after the selection of a service will be controlled by signalling associated with that service. The GEM

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

terminal shall monitor that signalling for changes made by the broadcaster. These changes may include the termination of particular applications as well as the addition of new auto-start applications.

Applications which are not identified as auto-start in the Application Description shall not be automatically launched by the GEM terminal, but require explicit launching. This explicit launching can be done by the resident navigator on the GEM terminal or by a GEM application. For example, the user can launch such applications after they have been offered a choice of applications through some user interface. Since the resident navigator is not required to provide a mechanism for this explicit launching, services wishing to have applications started must provide an application which is identified as auto-start.

Where the currently selected service in a service context includes multiple GEM applications, any running applications may be able to launch other applications from that set. The launched applications shall be presented inside that same service context.

A DVB-J application is able to achieve this using the application listing and launching API.

9.1.3 Support for execution of multiple simultaneous applications

The set of applications that are signalled within a service can be presented and executed concurrently.

GEM terminals shall be able to support applications from that set (using the same screen) at least as defined in annex G, "Minimum platform capabilities". GEM terminals are required to support execution of the set of such applications for each service which they permit to be presented simultaneously.

Broadcasters should ensure that simultaneous running of the set of applications for a service is comprehensible to the user and does not yield perceptible interference problems.

9.1.4 Stopping applications

GEM applications may stop themselves voluntarily using the GEM APIs or may be stopped by the GEM terminal in a number of situations. Examples of situations where this shall be allowed include:

9.1.4.1 A new service being selected replacing a previously selected one

When a new service is selected and replaces a previously selected one, applications from the former service shall only continue to execute where they are signalled in the new service. If an application is not signalled in that signalling then it will be stopped by the GEM terminal. Where an application is known to be bound to a single service, the broadcaster can identify that application as service bound using the `service_bound_flag` in the Application Description. Such applications shall be stopped as soon as possible by the GEM terminal and without needing the Application Description for the new service to be available. This allows the autostart application(s) of the new service to be started earlier than would otherwise be the case.

9.1.4.2 The stopping of an application by another application

Subject to the security policy of a GEM terminal, one application may request to stop another application. In such a case, the resident Application Manager, after a successful security check, kills the application; otherwise that application shall continue running, without interruption.

9.1.4.3 Changes in the application signalling to request a particular application be stopped

The broadcaster may request a GEM terminal to stop an application using the control codes in the Application Description. The precise semantics of these are dependent on the application format.

9.1.4.4 Stopping by the GEM terminal due to a shortage of resources

Where a GEM terminal has insufficient resources (e.g. memory, CPU and resources managed by the resource management API) to continue the execution of one of the running applications, the GEM terminal is allowed to decide to stop a GEM application without user intervention.

NOTE: The precise resources of a GEM terminal are implementation dependent.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 83 of 861

9.1.5 Persistence of Applications Across Service Boundaries

Where a running application is signalled in both the new service and the former service, and is not signalled as service bound in the former service, it shall continue to run and shall not be restarted. In this case, the running application shall become controlled by the application signalling of the new service where it is signalled and not the signalling of the former service. Hence the GEM terminal shall monitor the Application Description of the new service and shall stop responding to the Application Description of the former service.

If the application is signalled as service bound in the former service then it is terminated in the normal way as the new service is selected. If it is signalled as auto-start in the new service it will restart with no volatile context from the previous instantiation.

If an application survives a service selection operation, it will not automatically gain access to any new broadcast file system on the new service. It remains logically attached to any broadcast file systems it has already attached to, although it may be temporarily disconnected from the broadcast carousels feeding those file systems. The behaviour of the broadcast file system under these circumstances is defined in clause 6.2.5.3, "Loss of carousel behaviour".

If temporarily disconnected, the broadcast carousel shall be reconnected upon selection of a service where an identical carousel is available. This includes both the original service from which the application was downloaded (assuming the carousel is still present in that service) and other services containing an identical carousel.

9.1.6 Management of autostarting

The receiver shall launch autostart applications under the following conditions:

- The signalling indicates that the application can be supported by the receiver, as defined by the application profile and versioning information contained in the Application Description.
- Only a single application with a given Application identification is allowed to run at any time in a single service context.
- The application is a newly introduced autostart application or has newly been given autostart status.

So:

- When a service is selected the receiver shall launch at most one instance of each autostart application that it can support.
- If after service selection an autostart application that the receiver can support is introduced or a previously listed supportable application gains autostart status then the application shall be launched subject to normal resource limitations, etc.

However, if an autostart application terminates itself, it shall not be restarted unless it again becomes a new autostart application. An application becomes a new autostart application in the following cases:

- The receiver navigates away from the service and then selects a service where the application is autostart.
- The application is removed from the Application Description and then is re-introduced.
- The autostart status of the application is reset then set again.

NOTE 1: In summary the autostart status of an application is in effect an edge trigger rather than level trigger signal.

NOTE 2: These semantics for the autostart behaviour address "de-bouncing" the case where an autostart application terminates voluntarily. They do not address the case where the receiver terminates the application.

In this case the platform may attempt to re-start the application however this is implementation dependent.

NOTE 3: The present document does not describe in detail the timing required for the signalling to renew the autostart status of an application.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 84 of 861

9.1.7 Tuning without service selection

GEM applications may cause tuning to another transport stream by mechanisms other than service selection. Usage of these mechanisms does not constitute service selection and therefore no applications from the target transport stream or service shall be started either by the GEM terminal or by GEM applications. The GEM terminal shall continue monitoring the Application Description of the logically selected service where this is available on the target transport stream. Where the Application Description of the selected service is not available, the application shall continue executing as described in clause 10.4.2, "Visibility of Application Description and tuning". The service being presented in the service context shall not be changed by an application using these mechanisms:

- DVB-J tuning APIs.

This clause only applies to GEM terminal specifications with a tuner.

9.1.8 GEM Applications and Service Selection

GEM applications may select services using the service selection API. The service selection API includes a class `ServiceContext` to represent environments in which services may be presented. Calling the `select()` method on a `ServiceContext` causes a new service to be presented in that context and any former service being presented in that context will be stopped.

Where one GEM application uses the application listing and launching API to successfully start a second GEM application, the second GEM application shall be considered as executing inside the service context of the first GEM application. The number of `ServiceContext` objects representing a service context is implementation dependent, but each application sees only one such instance. Changes made by one application to its associated `ServiceContext` object are visible to the `ServiceContext` objects representing the same service context in other applications.

GEM applications started in response to a service selection operation are considered to be executing "inside" a service context. They may obtain a reference to the service context within which they are executing through using the method `getServiceContext(XletContext)` on `javax.tv.service.selection.ServiceContextFactory`.

9.1.9 Cached applications

When an application is signalled in the Application Description of a service in the usual way, and that Application Description entry also contains an `application_storage_descriptor`, the terminal may use files that have been stored on the terminal to accelerate the loading of that application. These files may have been stored on the terminal by any way allowed by clause 9.9, "Stored and cached applications", including proactive caching, or an explicit request to store or cache the application.

Stored broadcast related applications shall take the lifecycle model of service-bound applications. For example:

- The application can only be started when it is listed in the Application Description of the currently selected service.
- Following a service change, the application behaviour will depend on the Application Description signalling in the newly selected service.

In summary, the behaviour is that of a service-bound application, except that the loading is possibly accelerated due to the file system cache provided by the storage.

When a cached application (other than one with `launchable_completely_from_cache` set to "1") executes, it shall see the same filesystem as if it was being run directly from broadcast. The set of files stored in response to being listed in the Application Description file (see clause 9.6.1, "Applications loaded from the interaction channel") shall always override the same files from the broadcast.

Broadcasters should not remove a file listed in the ADF from the broadcast without amending the ADF and incrementing the application version number. If a cached file is removed from the broadcast, but is cached by the terminal, it shall still be readable. `File.list()` shall always return the current contents of the broadcast, so will not show such files.

When starting an application which has `launchable_completely_from_cache` set to "1", the behaviour depends on whether or not all critical files are cached. If they are not, the application will be started as described in the previous

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

paragraph (or, if "not_launchable_from_broadcast" is "1", the application will not be started at all). If all critical files are cached, the application will be started without mounting the carousel. Only the files stored in the cache shall be available. Note that, in this case, if the permission request file is not listed in the application description file (or does not have a priority of critical and has not been cached), the permission request file is missing and hence no additional permissions are requested.

NOTE: The application provider is responsible for managing any dependencies between those files which can be stored and those files which are not required to be stored.

9.1.9.1 Version management

If the application has not_launchable_from_broadcast set to '0' then the service-bound version shall be used regardless of the value of is_launchable_with_older_version. If the application has not_launchable_from_broadcast set to '1' and:

- is_launchable_with_older_version is set to '0', it shall not be launched at all.
- is_launchable_with_older_version is set to '1', the cached/stored version shall start.

Applications that use the application caching API are responsible for updating the cached version of an application when a new version is released. If not_launchable_from_broadcast is set to '1' and:

- is_launchable_with_older_version set to '0', a launchable application must be provided to manage the update (although a GEM terminal may perform the update autonomously).
- is_launchable_with_older_version set to '1', the cached/stored application is responsible to manage the update.

Terminals shall allow different versions of an application to be cached/stored simultaneously. (For example, if two broadcasters use the same application, they may not upgrade to the latest version at the same time.) Where multiple versions of an application are stored and one of them is to be started, the one started shall be the one whose version number is less than the version number of the transmitted application by the smallest amount.

NOTE: GEM terminals may refuse to update, or keep stored, applications where the version numbers change too frequently. Broadcasters should ensure that frequently changing files are not listed in the "Application description file". The meaning of "too frequently" is terminal-specific. However, an application version changing twice a day is not expected to happen too frequently.

When running a cached application where multiple versions of that application are cached, the terminal shall only return files from the cache that were stored in response to an application description file signalled with the same organisation ID, application ID, and version number as the version of the application that is being run.

9.1.9.2 Proactive caching

If a storable application is signalled, terminals are allowed to proactively store any files that are indicated in the ADF. Terminals do not have to honour the requested priority when proactively caching files.

In particular, note that proactive caching is not required to store all files with critical priority (but in this case the application shall not be reported as cached by the GEM APIs, and shall not be started if it is signalled with not_launchable_from_broadcast set to "1").

Terminals are responsible for handling version updates to applications that were proactively cached.

9.2 DVB-J Model

GEM terminal specifications based on the packaged media profile or the IPTV profile may not support the dynamic update of a service's Application Description. In such a case, corresponding language in clause 9 and its subclauses does not apply.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

9.2.1 Starting DVB-J Applications

DVB-J applications may be started by any of the means defined for general GEM applications. The application listing and launching API defined in annex S, "Application listing and launching" allows one GEM application to start another GEM application subject to security policy. The `start()` method of the `AppProxy` interface will then cause the Application Manager to start the new GEM application subject to normal resource limitations.

The Xlet interface is defined in the `javax.tv.xlet.Xlet` interface Java TV [16]. DVB-J applications provide a class implementing this interface and reference that class in the DVB-J application location descriptor. In order to start a DVB-J application, the application manager shall call the constructor of this class, the `initXlet()` and the `startXlet()` methods of this interface.

9.2.2 Stopping a DVB-J Application

DVB-J applications may stop for any of the reasons listed for general GEM applications. An application shall be able to notify that it is stopped by finishing its execution and informing the Application Manager through the `notifyDestroyed()` method on the `javax.tv.xlet.XletContext` interface. This interface also includes other methods to allow a DVB-J application to request or notify changes in its state.

The application listing and launching API allows an application to indirectly control the lifecycle of another application subject to security policy. This control is indirect because an application cannot invoke an Xlet state method directly, but goes through this API. This ensures that the resident Application Manager can always keep track of all the applications that are running.

When a DVB-J application is stopped by a GEM terminal, the `destroyXlet` method of the signalled Java class implementing the Xlet interface, i.e. the initial class of the application, shall be called by the application manager. In the case of the application being stopped due to a service selection operation, the stopping of the application shall be unconditional. This method call gives applications their last opportunity to save state before their execution stops. Applications which wish to survive the user of the GEM terminal zapping away from their service (e.g. during an advertising break) must save their state and reload that state when they are re-started if the user returns to that service later.

9.2.3 DVB-J Application Lifecycle

9.2.3.1 Introduction

This clause describes the Xlet lifecycle model for the DVB-J API. This describes the capabilities of the Xlet in each state and the methods by which the application manager influences the life cycle state.This clause is not directly related to other aspects of a system, such as graphics or shared resource allocation/management.

NOTE: The Java platforms define a number of application models that have their own lifecycles associated with them. In general, they are designed to address specific issues on that platform. For instance, the Applet was designed to provide support for executable content in web pages. However, none of the existing application technology fully addresses the specific requirements of television receivers. The application lifecycle defined in this clause is meant to be compatible with existing Java platforms and virtual machine technology.

This clause defines the lifecycle of an instance of a DVB-J application. The lifecycle of a DVB-J application and of an application instance are the same except that when an application instance is destroyed, the application is only transiently destroyed and then becomes not loaded. See `org.dvb.application.AppProxy.NOT_LOADED` in annex S, "Application listing and launching".

9.2.3.2 Lifecycle state machine for DVB-J application instances

The Xlet state machine ensures that the behaviour of an Xlet is close to the behaviour that television viewers expect, specifically:

- The perceived start-up latency of an Xlet can be very short.
- It is possible to put an Xlet into a state where it is not providing its service.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- It is possible to destroy an Xlet at any time.

The figure 13 shows the state machine model for Xlets. The Xlet states are defined in more detail in table 8.

The different influences that can cause an Xlet to change state include:

- The application manager uses the Xlet API to signal these changes to the Xlet.

  Various factors may stimulate the application manager to act in this way, for example:

  - Broadcast signalling (e.g. a change in the state of the application_control_code parameter carried by the Application Description (see clause 10.4, "Application Description")).

  - User selection of an application in a host provided UI.

- The Xlet itself "decides" to change state.

  The application uses the `XletContext` Object to communicate or request such changes to the application manager.

- Another Xlet acts via the application launching API (see clause 11.7.2, "Application discovery and launching APIs").

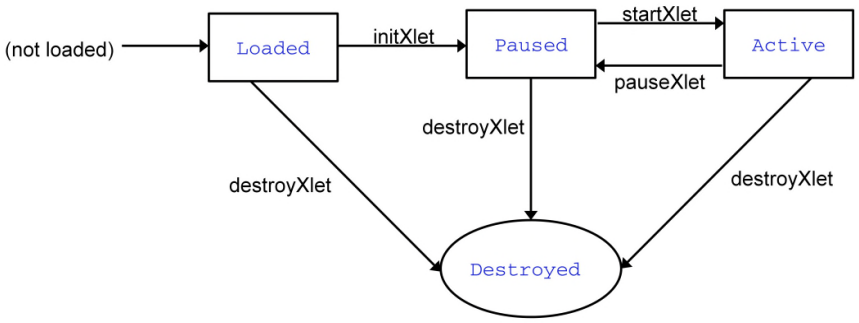


Figure 13: Xlet lifecycle state machine diagram

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 8: Valid lifecycle states of DVB-J application instances

State Name	Description
`Loaded`	The DVB-J application instance has been loaded and has not been initialized. The signalled Java class used to initiate a DVB-J application instance must implement the `javax.tv.xlet.Xlet` interface. Otherwise, the class (and hence the application instance) may be ignored. An instance of the signalled Java class is created by the application manager e.g. using the `Class.newInstance` method. Therefore a DVB-J application instance must have a public "default constructor". Otherwise, the class (and hence the application instance) shall immediately enter the `Destroyed` state. If the default constructor returns without throwing an uncaught exception, then the application instance is considered to be "`Loaded`", otherwise the application instance immediately enters the `Destroyed` state and is discarded. Once the application instance has been successfully loaded and instantiated, the application manager can transition the application instance to the Paused state by invoking the `initXlet` method on the signalled class (implementing the Xlet interface). If the `initXlet()` method throws an `XletStateChangeException` then the application instance shall remain in the Loaded state. The only possible state transition for such an application instance is into the destroyed state. The application instance can request this itself or wait for the application manager to cause this transition. (See notes 1 and 2.)
`Paused`	A Paused DVB-J application instance should minimize its usage of resources if it wants to maximize its probability of survival. This does not imply that it cannot be holding any resources, but in such a case, it would have a lower priority as concerns access to resources than it had when it was in the `Active` state. When entering the paused state, an application should ensure that any `HScene` instances it is using are invisible. See also clause 9.10, "Lifecycle interactions between GEM and resident applications". While an application is paused, it should not attempt to make any `HScene` instances visible, and it should not attempt to draw to the screen. Any application that does this while paused may be considered uncooperative, and may be terminated by the application manager. Applications should also respect the rules of "television viewing mode" as defined in clause 11.4.1.4. This state is entered: • From the Loaded state after the `Xlet.initXlet()` method returns successfully when invoked by the application manager (the first time). Other invocations of this method do not cause the change of state. (See note 3) • From the `Active` state after the `Xlet.pauseXlet()` method returns successfully when invoked by the application manager. Other invocations of this method do not cause the change of state. • From the `Active` state upon entering the `XletContext.notifyPaused()` method.
`Active`	The DVB-J application instance is functioning normally and providing service. This state is entered: • From the Paused state after the Xlet.startXlet() method returns successfully.
`Destroyed`	The DVB-J application instance has released all of its resources and terminated. This state is entered: • When the Xlet's destroyXlet() method returns successfully. The destroyXlet() method shall release all resources held and perform any necessary clean up so the Xlet may be garbage collected. • Upon entering the XletContext.notifyDestroyed() method. The Xlet performs its clean up actions before calling the notifyDestroyed() method. (See note 4)
NOTE 1: Application initialization is intended to occur in the initXlet method, rather than in the default constructor. NOTE 2: This state is entered only once per instance of an DVB-J application. NOTE 3: The application manager shall only call initXlet() once per instance of a DVB-J application. NOTE 4: This state is entered only once per instance of an DVB-J application instance.	

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Every time a DVB-J application instance is started (i.e. the constructor of the object implementing Xlet is called), it shall logically run in its own new virtual machine instance. See clause 11.2.1, "Basic Considerations".

Only the DVB-J application can determine if it is able to provide the service for which it was designed. As such, in some respects an application manager cannot guarantee whether an DVB-J application can, or is, providing its service; and application manager can only indicate that the DVB-J application is able to do so. A typical sequence of DVB-J application execution is:

Application Manager	DVB-J application
The application manager creates a new instance of an Xlet.	The Xlet's default (no argument) constructor is called, it is in the `Loaded` state.
The application manager creates the necessary context object for the DVB-J application to run, and initializes the Xlet.	The DVB-J application uses the context object to initialize itself. It is now in the `Paused` state.
The application manager has decided that it is an appropriate time for the DVB-J application to perform its service, so it signals it to enter the `Active` state.	The DVB-J application acquires any resources it needs and begins to perform its service.
The application manager no longer needs the DVB-J application to perform its service, so it signals the DVB-J application to stop performing its service.	The DVB-J application stops performing its service and might choose to release some resources it currently holds.
The application manager has determined that the DVB-J application is no longer needed, or perhaps needs to make room for a higher priority application in memory, so it signals the DVB-J application that it is a candidate to be destroyed.	If it has been designed to do so, the DVB-J application saves state or user preferences and performs clean up.

9.2.4 Xlet API

The Xlet API provides GEM application developers with an API that provides life cycle signalling. The Xlet API uses the callback approach to signal state changes.

This API is specified in clause 11.7.1, "APIs to support DVB-J application lifecycle".

9.2.4.1 Xlet State Change Semantics

An Xlet's state can change either by having one of the methods on its Xlet Interface called, or by making an internal state transition and notifying the application manager via the `XletContext` Object. The semantics of when that state change actually happens are important:

- Calls to `Xlet`: this interface indicates a successful state change only when the call successfully returns.
- Calls to `XletContext`: the `notifyDestroyed()` and `notifyPaused()` methods indicate a state change on entry. The `resumeRequest()` method indicates no state change, instead only a request to change state.

If a method on the Xlet interface throws an `XletStateChangeException`, the Xlet shall remain in the state it was in immediately prior to the call of the method throwing the exception unless otherwise specified. In the present document, the only exception to this rule is the `destroyXlet` method when the `unconditional` flag is `true` where throwing the `XletStateChangeException` is specified to have no effect. For the case of `initXlet` which may only be called once, the application manager may choose to transition the Xlet to the destroyed state (without calling `destroyXlet`) some implementation specific time later.

9.2.4.2 Xlet state change requests

The following table defines the previous states in which calls to the methods on `XletContext` relating to state management are valid;

NOTE: These methods are called after the Xlet has changed its state.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 9: States for valid state management calls

Call	State
notifyDestroyed	all states
notifyPaused	active only
resumeRequest	paused only

Calls to these methods when an Xlet is in any other state shall have no effect.

9.2.5 Multiple application environment support

The DVB-J platform allows for the simultaneous execution of several DVB-J applications.

Allowing several DVB-J applications to run simultaneously implies that some rules be defined for these DVB-J applications to share the resources of the MHP, and in particular for them to share the Input Focus and the Output Focus.

9.2.5.1 Control of DVB-J applications by other DVB-J applications

The present document provides support for control of the lifetime of a DVB-J application by another DVB-J application. This feature enables service providers to write their own "Launcher applications" that take care of the presentation to the user of the availability of DVB-J applications, and that enables eventually the user to launch DVB-J applications.

NOTE: The actual control of the lifetime of an DVB-J applications is done by the Application Manager only. The present document only provides APIs that enable DVB-J applications to ask the Application Manager to start, stop, pause and resume DVB-J applications.

See clause 11.7.2, "Application discovery and launching APIs".

9.2.5.2 Input Focus management

The input focus is defined as follows:

- The application that has input focus is in principle able to receive user-input events.
- Other applications not having the input focus can request to receive a subset of user-input events via a dedicated API. See clause 11.3.2.2, "org.dvb.event".

A DVB-J application has input focus if and only if the `java.awt.Component` having focus belongs to the component tree of that application.

9.2.5.3 Other resources management

The APIs defined in the present document provide support for resource allocation/revocation and resource revocation notification. The semantics of the APIs, however does not define under which circumstances an access to a resource is granted or revoked. While it is well understood that in most cases, it is up to the GEM implementation to define its own policy in terms of resource management, this clause defines the basic rules that a GEM implementation has to follow.

The GEM specification describes a multi-application environment. Hence several applications may be competing for access to the same atomic resource. The resource notification API described in clause 11.7.5, "Resource notification" provides a common way for applications to negotiate access to scarce atomic resources when such competition happens. This API allows for the GEM terminal to inform the application that currently holds the resource that another application wants to access this resource. It also provides a means for the owner of the resource and to the requester of the resource to communicate by private means. This private communication is reflected by the `request_data` object that the requester may pass to the owner. The semantics of this object is private and has to be known by both applications.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Some existing and general purposes java APIs that were developed before the GEM work was started do not use this general resource sharing mechanism. Hence access to resources addressed by these APIs are not subject to negotiation. For example, when an application holds a JMF player, if another application was to create a JMF player for the same content-type, the GEM terminal has to decide by itself whether it withdraws the resource underlying the JMF player from the current owner and grants it to the requester.

It is also possible for applications to use the inter-application communication API to establish private communication channels enabling them to negotiate access to resources.

As an optimisation, GEM implementations may make available additional resources (e.g. tuners) to GEM applications which do not ask for them explicitly. Any such additional resources must be successfully reserved on behalf of the application before use and released afterwards. If the reservation fails, the same resource shall be used as on a GEM implementation without those additional resources, e.g. the tuner associated with the service context in which the application is running. If the additional resources are later removed, the GEM implementation shall revert to using the expected resources (e.g. the tuner associated with the service context in which the application is running) transparently to the application.

9.2.5.4 VM implementation

Where there are multiple DVB-J applications being executed as part of the same service, GEM terminals are allowed implement these in a single actual virtual machine instance. Regardless this shall conform to clause 11.2.1, "Basic Considerations".

9.3 Void

9.4 Inter-application resource management

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

9.4.1 Application instances running in the same service context

Where there is a resource conflict between two applications signalled as part of the same service and running in the same service context, this shall be resolved using the priority signalled in the `application_priority` field in the Application Description for each application. When comparing two applications, the one with the higher `application_priority` value shall be considered the more important to preserve.

- Where there is a resource conflict between two applications signalled as part of the same service and running in the same service context, this shall be resolved using the priority signalled in the `application_priority` field in the Application Description for each application or the external application authorization descriptor depending on which descriptor the application is currently signalled by.
- Unless stated otherwise, only the application which owns a resource has the right to modify the state / settings / configuration of that resource. If a resident application (e.g. the navigator) makes changes to the state / settings / configuration of a resource, the GEM terminal shall inform the GEM application which formerly owned the resource that the GEM application has lost ownership of the resource.

NOTE: The only exception to this rule is where all applications running inside a service context together own the service component handlers of that service context and not any single one of them. See clause 11.6.2, "Service selection API".

When a `ServiceContext` is in the presenting state, it shall keep reserved the resources needed to present the service currently being presented. In particular, when the service being presented is received via a tuner represented by a `org.davic.net.tuning.NetworkInterface`, that tuner shall be kept reserved. It is implementation dependent whether such tuners are reserved when not being used for the presentation of the currently selected service. Any resources reserved by the implementation of `ServiceContext` but not used for the presentation of the currently selected service shall be released if any other application requests them.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

If an application running inside a `ServiceContext` attempts to reserve a resource being used for presenting the currently selected service in that `ServiceContext`, it shall be given a higher priority than the implementation of `ServiceContext`. e.g. an application running in a TV service running in a `ServiceContext` shall be able to reserve the `NetworkInterface` currently used to present that TV service and tune away to another transport stream.

9.4.2 Application instances not running in the same service context

The present document intentionally does not specify resource management policy or algorithms other than as above. The service level resource priority (as set by the `setPriority()` method on `UnboundApplicationService` or `DvbServiceContext.setPriority()`) is one input to the resource management algorithm. Another input may be the identity of the application or applications that the user is interacting with. How these inputs are used by the implementation resource management algorithm is not defined by the present document.

NOTE: Specifically the present document intentionally does not require support for the platform-wide application resource priority defined in clause 10.2.2.5 "Application Priority" of OCAP [5] or the policies and mechanisms defined in clause 19 of OCAP [5].

9.5 Void

9.6 Services and applications not related to conventional services

Clause 9.1, "Service-bound GEM applications" describes applications whose lifecycle is controlled through services. The present document additionally supports applications whose lifecycle is not bound to that of services. These include:

- Applications signalled through the interaction channel (see clause 9.6.1, "Applications loaded from the interaction channel").
- Applications running from storage in the GEM terminal (see clause 9.6.2, "Stored services").

9.6.1 Applications loaded from the interaction channel

In this scenario the application signalling comes from the interaction channel rather than the broadcast channel. In overview:

- Data equivalent to the service-bound Application Description is provided from the interaction channel in the AIT File. The connection to the server that provides the data is analogous to a connection to a service.
- A locator specifying the location of the AIT File is analogous to the locator specifying a service.
- Navigation to this locator is analogous to selection of a service. The service selection could either be from the GEM terminal's intrinsic navigator or via an application using the service selection API.
- As with navigation between services when an interaction channel locator is selected as the current service the GEM terminal retrieves and decodes the AIT File. Decisions about which currently running applications are allowed to remain running and which new applications are launched are based on the information in the AIT File just as they would be when navigating between services.
- When a locator specifying an AIT File is successfully used in a service selection operation, the applications database shall be populated with the information from that AIT File.

9.6.2 Stored services

Stored services are an encapsulation of one or more applications. The lifecycle of a stored service shall obey the same rules as defined for a service in clause 9.1, "Service-bound GEM applications". So, for example, currently running applications will be stopped if they are not listed to be runable in the stored service's Application Description information.

Hence there is no requirement to execute a stored services simultaneously with a service or application.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Stored applications running as part of a stored service shall:

- Execute with the Application identification that was stored with them when they were stored.
- Be able to access a file system using GEM APIs where:
  - Any file that is present shall have the same name, relative path as it did in the broadcast file system.
  - Any API providing byte level access to the data of a file shall return the same data as was broadcast.
  - The set stored may be a superset of the files listed in the Application description file.
  - That file system is read-only to the GEM application.

NOTE 1: This does not preclude the implementation also storing an optimized representation of files that would be presented to certain APIs. For example, this allows class files to be precompiled for subsequent presentation at the class loader.

The results of an application attempting to access files or directories above the directory that originally contained the Application description file are implementation dependent but shall either succeed or fail using the failure modes specified for the mechanism used.

When a stored service is started, any previously running streamed media decoders shall continue to run. It is the responsibility of applications to stop them if so required. Any resources used by such streamed media decoders shall have the lowest possible priority in any resource conflicts.

A stored service has the same lifecycle semantics as other services types (such as broadcast). So, when a stored service is selected the stored Application Description information is inspected to determine which (if any) applications should auto start and which (if any) currently running applications are allowed to continue running by virtue of being included in the stored service.

NOTE 2: Currently, no API has been defined that would allow the addition of externally authorized applications to a stored service.

A stored service is built up by identifying applications from another service. It is not a copy of a service.

When an application is added to a stored service, the following information shall be stored:

- The storage_property and version information from the Application storage descriptor (see clause 10.8.2, "Application storage descriptor").
- Sufficient information to be able to reconstruct the application_descriptors_loop (e.g. so that it can be returned by the `org.dvb.application.AppAttributes.getProperty()` method).
- The application type.
- The Application identification.
- At least all those files listed in the Application description file with priority of "critical".
- Implementations may store files in addition to those listed in the Application description file.
- Sufficient information to be able to construct the set of permissions to be granted to the application including the certificate files needed for any credentials.

Implications of the above are:

- If any non-critical files are needed for the application functionality then the application is responsible for obtaining them in the event that they are not stored.
- No information from Application Description common loop descriptors is required to be stored.
- Information from the transport protocol descriptors is only relevant during the storing of the application. When the application is executing from storage it is responsible for creating the transport connections it requires.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 94 of 861

If applications within a stored service are also listed in a currently selected service they are runable with the same constraints as other broadcast applications in the context of that service. So, as with "Cached applications", the consequence of storage is to allow better performance.

If a stored application service is removed while it is being presented then presentation of that service will stop with a `PresentationTerminatedEvent.SERVICE_VANISHED`. The navigator may select another service in an implementation-dependent manner. (For example, a stored application may choose to remove the service it is stored in, including removing itself. This is not an error.)

Stored application services which contain no autostart applications should not be presented to the end-user, either by the navigator or by GEM applications such as EPGs. GEM terminals should delete stored application services which contain no applications when/if no running GEM applications have references to that service.

9.6.3 DVB-J Model

This clause is applicable only if clause 9.6.1 or 9.6.2 is required by the GEM terminal specification.

For DVB-J applications, APIs required to include some support for these applications include the following.

- Service selection API (see clause 11.6.2).
- "Content referencing" (see clause 11.7.6) as described in clause 11.11.12, "Support for the HTTP Protocol in DVB-J".
- "Protocol independent SI API" (see clause 11.6.5) as described in clause 11.11.12, "Support for the HTTP Protocol in DVB-J".

When a stored service is successfully created using the method `StoredApplicationServiceFactory.createStoredApplicationService`, this service shall appear in the list of services maintained by the `SIManager`. It shall be returned by `filterServices` both when passed an instance of `ServiceTypeFilter` constructed with the type `StoredApplicationServiceType.STORED_APPLICATION_SERVICE` and when passed null to list all known services. The method `ServiceContext.select` shall support selecting stored services as identified by instances of `StoredApplicationService` created or returned using all these mechanisms. This is described in more detail in clause 11.12.2.2, "Modified behaviour of GEM 1.0 APIs".

In profiles where application signalling over the interaction channel is supported, the method `SIManager.getService` shall accept instances of `javax.tv.locator.Locator` whose external form is valid "http" and "https" URLs and return a `Service` object. The method `ServiceContext.select` shall support selecting services identified by such service objects. This is described in more detail in clause 11.11.12, "Support for the HTTP Protocol in DVB-J".

9.6.4 Common behaviour

This clause is applicable only if clause 9.6.1 or 9.6.2 is required by the GEM terminal specification.

The environment in which stand-alone GEM applications are presented or executed shall be the same as for applications related to a conventional service except as detailed below.

- Any existing video and audio running from before the application was started shall continue running.
- Applications shall not have a mechanism for attaining the identity of any such existing video or audio.
- Applications wishing to hide any such existing video may cover the full screen area with graphics.
- Applications have no ability to hide any such existing audio.
- GEM terminals shall assign the lowest possible resource priority to the presentation of any such existing video and audio.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

9.7 Lifecycle of internet access applications

9.7.1 General issues

GEM terminals supportinginternet access applications shall support at least the execution of one internet access application at one time. There is no requirement to support the execution of more than one internet access application at one time. There is no requirement to support the execution of internet access applications and GEM applications at the same time. GEM terminals where internet access applications and GEM applications can be in memory and executing code at the same time but cannot share access to the screen are allowed. Such GEM terminals shall be considered as supporting internet access applications and GEM applications at the same time in the rest of the present document. In these GEM terminals, any GEM applications in the active state may be put in the paused state when they lose access to the screen to an internet access application. The present document is intentionally silent about priorities for access to resources like memory in the event of a conflict between GEM applications and internet access applications.

NOTE: This means that when running internet access applications and GEM applications simultaneously, if memory in the GEM receiver runs low, some implementations may choose to terminate the GEM applications and others may choose to terminate the internet access applications.

9.7.2 Starting internet access applications from GEM applications

Where GEM terminals support simultaneous execution of internet access applications and GEM applications at the same time, then the internet access application shall start when requested. The internet access application shall be considered to be a peer of the GEM application and not a child of it. The lifecycle of the internet access application is not constrained by the lifecycle of the GEM application which started it.

Where GEM terminals do not support this simultaneous execution then the GEM application shall be destroyed / killed as part of the starting of the internet access application. In this case, when the end user of the GEM terminal exits the internet access application for reasons other than selecting a "dvb:" link to a service, the GEM terminal shall return to presenting the service (if any) which it was presenting before the internet access application was started. Any GEM applications in this service will be re-started without any previous state using the normal mechanisms defined in the present document.

The APIs to enable DVB-J applications to manipulate internet access applications are defined in annex AH, "Internet client APIs".

9.7.3 Selecting DVB services from internet access applications

In GEM, internet access applications shall support end user starting of DVB services (i.e. using the 'dvb:' URL) and GEM applications signalled over the interaction channel (i.e. using an application description file referenced by an HTTP URL). This shall be handled by the normal service selection mechanism used elsewhere in the GEM specification (e.g. in the implementation of the service selection API).

On GEM terminals which support running internet access applications and GEM services at the same time, the new GEM service shall start running if and only if the service context in which the internet access application is running can support the running of GEM services. If the service context in which the internet access application is running cannot support the running of GEM services then the service selection operation shall fail and this will be reported to any GEM applications listening for service selection events on that service context. If the internet access application was started by a GEM application using the `org.dvb.internet` API, that GEM application is still running, and that GEM application has registered for `InternetClientEvents`, then if the service selection fails, that application will also receive an `InternetClientFailureEvent` including the locator of the DVB service which failed to be selected.

NOTE: It is up to the GEM application receiving an `InternetClientFailureEvent` to extract the locator for the DVB service which failed to be selected and, if it so desires, select it in its own service context. If it doesn't do this then the end-user's starting of the DVB service fails.

If the internet access application was started directly from the navigator and not by a GEM application, the service context used to present any DVB service selected is implementation dependent but shall be one which can support the selection of DVB services.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

On GEM terminals which do not support running internet access applications and GEM applications at the same time, the internet access application shall behave as if the end user of the GEM terminal had asked to exit that application. This may result in a platform specific dialogue with the end user, e.g. to ask about saving partly composed email messages as a draft. Depending on the nature of any such dialogue, the selection of the DVB service may fail if the end user of the GEM terminal does not wish to exit the internet access application concerned.

There is no requirement to remember the internet access application which selected a DVB service or to return to that internet access application when leaving the DVB service which it selected. This is the opposite of starting internet access applications from GEM applications.

9.8 Plug-ins

Attention is drawn to the general rules in clause 4.2.

Clause 5.4, "Plug-ins" of the present document defines the concept of plug-ins for use in migration situations, for the support of applications in existing content formats. Services and networks using such an approach to migration have a number of choices defined by the present document. Where a service carries one or more applications in an existing content format (referred to as delegated applications), those applications may be signalled either using the native signalling defined for that existing content format or using the Application Description based signalling defined in the present document or both. Two types of plug-ins are defined: inter-operable plug-ins and implementation specific-plug-ins. The present document is intentionally silent about the operation and signalling of implementation specific plug-ins. Signalling of inter-operable plug-ins shall be done as defined in clause 10.7.2, "GEM signalling scenario".

The lifecycle of inter-operable plug-ins shall obey the semantics of DVB-J applications concerning service selection operations as defined in clause 9.1.4.1, "A new service being selected replacing a previously selected one" of the present document.

Where the Application Description is used to signal delegated applications, the maintainer of the specification for that format shall register with the DVB project to obtain a value for the "application type" field of the Application Description (see table 12, "Application types"). Having done this, the maintainer of the specification for the existing content format is responsible for defining any additional descriptors for the Application Description which are required for the support of their content format.

9.9 Stored and cached applications

This section describes the common storage behaviour that applies to all stored applications. See clause 9.1.9, "Cached applications" and clause 9.6.2, "Stored services".

9.9.1 Storing files

When a file that forms part of an application is stored, the following metadata shall be stored with the file:

- The application type.
- The Application identification.
- The version information from the Application storage descriptor (see clause 10.8.2, "Application storage descriptor").
- If the Application description file lists any GEM security file (such as a signature file or a CRL) then these files shall be stored in the normal way (with due regard to the priority, etc.). However, it is not necessary to list the security files in order to retain the security information as the method of retention of this information is an implementation detail.
- Sufficient information about the certificate chain that authenticated the application to be able to determine if any of those certificates have been revoked when it comes to run the application.
- Sufficient information to fully implement the method `DSMCCObject.getSigners` for all signed files which are stored.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 97 of 861

Implications of the above are:

- It is not required to store file MIME type as carried by the optional Content type descriptor. The application developer is responsible for ensuring that stored files can be correctly interpreted without relying on this information.
- It is not required to store cache priority information from the Caching priority descriptor. The application developer is responsible for ensuring that dynamic data is not listed in the Application description file.

9.9.2 Version management

To be eligible for any kind of application storage, an application must be signalled with an `application_storage_descriptor`. This contains the version number for the application.

A stored application is uniquely identified by an `organisation_id`, `application_id`, version number triple. Terminals shall support storing multiple versions of the same application. (For example, different service providers may use the same third-party application but may update to a new version at different times.)

Application authors are responsible for ensuring that their application is always broadcast with the correct version number.

If the broadcast version number of an application changes whilst that application is being stored or cached (including as part of pro-active caching by the terminal) any files stored since the last time the Application Description was received with the old version number shall be erased. If this storing or caching was requested via the GEM APIs and is not pro-active caching, the first time this happens, the platform shall automatically attempt to store or cache the new version of the application instead of the old version. If the broadcast version changes again whilst the same application is still being stored/cached, the download shall fail with an `ApplicationDownloadException`.

9.9.3 Removing stored applications

If an application is running (i.e. in any state other than `DESTROYED` or `NOT_LOADED`), the terminal shall not remove it from storage or cache.

The API methods to remove an application from storage or cache shall work normally and shall not fail just because the application is running. When the application terminates, only then shall it be removed from storage or cache.

For applications that are running as part of a stored service, removing them from that stored service shall cause them to be stopped as if they were removed from the Application Description (i.e. the `destroyXlet()` method shall be called as normal, and that `destroyXlet()` method shall be able to access stored files).

9.9.4 Interrupted downloads

Storing or caching an application may be interrupted, e.g. if the application that requested the storing or caching terminates or is terminated. (For example, if the user zaps to a service whilst a download is in progress.)

If storing or caching an application is interrupted before all critical files are downloaded, terminals are recommended to keep the partially-downloaded application stored in an implementation-specific cache, and if the terminal is asked to download the same version of that application again, then the terminal should resume where it left off and not download the already-stored files again.

If storing or caching an application is interrupted after all critical files are downloaded, the storing or caching has succeeded, even if there are non-critical files which could still be downloaded and have not been.

9.9.5 Dynamic behaviour

If an application that was not previously stored becomes stored, the following shall apply:

- An `AppsDatabaseEvent` with event id `APP_ADDED` shall be sent for that application.
- If the application was signalled in the current service as both `AUTOSTART` and `not_launchable_from_broadcast`, it shall be started.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- Subsequent calls to `AppsDatabase.getAppAttributes` for that application will return an instance of `ExtendedAppAttributes` whose `isStored()` method will return true until/unless the application is later removed from storage.

NOTE: For applications with `not_launchable_from_broadcast` set, the return value of the `AppAttributes.isStartable()` method will change at the same time.

9.10 Lifecycle interactions between GEM and resident applications

Application authors should be aware that applications may be made invisible without having other resources withdrawn, and may not be put in the paused state.

9.11 Providers

9.11.1 Introduction (informative)

Providers is a term used in Java SE for extensions to platform facilities exposed via standardised APIs. These are often adaptors between implementations of standard APIs and market specific protocols. They use standard APIs and are independent of any particular implementation and hence can be deployed by operators directly without a system software update or similar process.

The present document defines a framework for these. Example providers enabled by that framework include the following:

- provider for the PKCS #11 protocol for connecting smart cards to return channel security;
- providers for connecting the standard GEM SI APIs to proprietary SI formats and protocols;
- providers for connecting the standard GEM media presentation APIs to various protocols for control of media presentation in IPTV, both proprietary and variations on standards.

Providers may be restricted in scope to a single GEM application or may apply to all GEM applications in a GEM terminal. The first of these (called Xlet bound) are available to normal signed GEM applications. The second of these (called System bound) are only available to privileged applications.

Three models for provider class distribution are supported:

- Provider classes are distributed as part of the application using them. They are loaded by the application's Classloader and without any special extension to the signalled classpath. The provider classes need not be stored in the GEM terminal. Each application using the provider has its own copy of the provider classes.
- Provider classes are distributed as part of an application. This application declares (via signalling) that it contains the classes for that provider. When an application using that provider starts, the classes for the provider are used from the application that contains it without an instance of the containing application being started. These provider classes may be loaded from the network or from storage if the containing application has been stored, e.g. using the mechanisms defined in clause 9.9, "Stored and cached applications" of the present document.

  They are loaded either by the application's Classloader (with an extended classpath) or by a parent of that Classloader not shared by any other application. Each application using the provider has its own copy of the provider classes.

NOTE: The application containing the provider classes is used as a container for those classes and may never be instantiated as an application instance in its own right.

Such applications may be signalled with application control code `DISABLED`. `StoredApplicationServices` which only contain applications signalled with this application control code will not be selectable as defined by `StoredApplicationService.isSelectable()`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- Provider classes are distributed as part of a different application from the one(s) using them. They are installed by an instance of that application. They are loaded by the Classloader of that application. They are never visible to any other application. There is a single copy of those classes.

The first two of these models are only used for `XletBoundProviders` and the third is only used for `SystemBoundProviders`.

9.11.2 Lifecycle of xlet bound providers

When an instance of an application with one or more provider usage descriptors in its Application Description starts, the implementation shall attempt to find a provider for each signalled provider name by matching the provider names. For each matching provider, the following shall be done:

- The classes of the application containing the provider shall be added to the classpath of the application being started (or added to the classpath of a parent classloader of the application being started which is not shared with any other application).

- An instance of the class whose name is signalled in the `provider_class_name` shall be created and registered with the `ProviderRegistry` as an `XletBoundProvider`. This shall be created before the constructor of the Xlet class is called.

If no provider is found for a signalled provider name, the implementation shall make provision in the classloader of the application instance in case a provider of that name is installed during the lifetime of that application instance. If a provider of that name is installed during the lifetime of that application instance, the steps listed above shall be followed and the provider shall become available to the application.

Providers may be registered and unregistered during the lifetime of an application instance however any providers which are still registered when the application instance terminates shall be unregistered at that time.

While any application instance is using a provider exported by a second application, that second application shall not be removed from storage, see clause 9.9.3, "Removing stored applications". If a request to remove the second application succeeds while providers it exports are in use, it shall only be removed once all of the application instances using providers it exports have terminated. Once such a request has succeeded, no new providers shall be installed from the removed application. If providers of that name are to be used, a new application must be stored which exports that a provider of that name. Once such a new application has been stored, that application shall be used for new providers even though the former application may not actually have been removed yet.

9.11.3 Lifecycle of system bound providers

These are registered by the application containing the provider. The Xlet which registered the provider may unregister it. Otherwise the provider will be unregistered when the Xlet which contains it terminates.

NOTE: Xlets containing system bound providers would typically be unbound applications without any user interface.

9.12 Impact of graphics constraints on the application model

9.12.1 Impact on generic applications

The graphics constraints descriptor (clause 10.4.3.3, "Graphics constraints descriptor") shall be used in the following ways:

a) Selecting the default graphics configuration for an application when an instance of that application starts or deciding not to start the application.

b) Defining whether an application can continue to show a user interface (or continue to run) when the set of available graphics configurations changes.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

When an instance of an application starts, its requested graphics configuration(s) shall be compared against the available graphics configurations for the context in which the application is being started.

- If exactly one graphics configuration requested by the application is available then that one shall become the default graphics configuration of the application.
- If more than one graphics configuration requested by the application is available then the most preferred available configuration shall become the default graphics configuration of the application.
- If no graphics configuration requested by the application is available then the application instance shall either run without a visible UI or not run at all depending on the value of the `can_run_without_visible_ui` flag.

While an application instance is running, if the set of available graphics configurations changes then the following shall apply:

- Running applications signalled with `handles_configuration_changed` set to 0 shall be killed if their default graphics configuration is not available after the change.
- Running application signalled with handles_configuration_changed set to 1 where none of the requested graphics configurations are available shall either be killed or run without a visible UI according to the value of their `can_run_without_visible_ui` flag.
- Running applications signalled with `handles_configuration_changed` set to 1 where at least one of the requested graphics configurations are available shall run with their default graphics configuration being the most preferred which is available after the change. If this is different from the graphics configuration used before the change then this may result a change in the size and/or the HGraphicsDevice of their HScene.

Table 10 specifies the behaviour of applications in the same service as video when an application external to the service context where the video is presented takes control of video presentation.

Table 10: Application behaviour under external control of video presentation

handles_externally_controlled_video	can_run_without_visible_ui	Video already under external control when application starts	Application already running when video comes under external control
0	0	Fails to start	Killed except for applications with no visible UI which continue to run.
0	1	Starts but shall be unable to obtain a visible UI	UI becomes invisible
1	0	Starts with visible UI	Continues to run
1	1	Starts with visible UI	Continues to run

9.12.2 Impact on DVB-J applications

The default graphics configuration of the application shall be used for the HScene returned by the implementation of the following methods:

- `org.havi.ui.HSceneFactory.getDefaultHScene()`
- `org.havi.ui.HSceneFactory.getDefaultHScene(HScreen)`
- `javax.tv.graphics.TVContainer.getRootContainer()`
- `javax.microedition.xlet.XletContext.getContainer()`

NOTE: Regardless of the signalled default graphics configuration, applications may use the other facilities provided by the `org.havi.ui` package to discover HGraphicsConfigurations and attempt to select them subject to being able to reserve the corresponding resource.

For a DVB-J application, running without a visible UI shall mean the following:

- Already running applications shall have the visibility of their HScene set to `false`. Attempts to change this shall fail silently while the application remains running without a visible UI.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- Applications starting without a visible UI shall have any attempts to obtain a default HScene fail for methods with a defined failure mode. For methods without a defined failure mode, an HScene shall be returned but attempts to set the HScene visibility to true or to call the show method shall fail silently.

In table 10, "Application behaviour under external control of video presentation" DVB-J applications shall be considered to have no UI if they do not have any HScene instances except for ones which have been disposed.

9.13 Unbound Applications

This clause is optional in all profiles of GEM. GEM terminal specifications may include all or part of the specification elements of this clause, or may include none at all. If specification elements are included, functional equivalents may be specified where necessary.

9.13.1 Introduction to unbound applications (informative)

9.13.1.1 Scope

Unbound applications are ones whose lifecycle is not bound to the presentation of a particular piece of content, either a service or content provided on demand. This is in contrast to broadcast GEM applications (see clause 9.1, "Service-bound GEM applications") which are bound to one or more TV channels by signalling and only run while that TV channel is being presented. The effect of unbound applications can be achieved in GEM 1.0 by signalling them as part of all TV channels in a network with the result that they run regardless of which TV channel or service is selected.

OCAP [5] adds support for 3 different versions of unbound applications:

- Host device manufacturer applications installed with the OCAP implementation.
- Applications that are signalled through the XAIT.
- Applications that are registered through the Monitor Application.

The second and third of these are in the scope of the present document. The first of these is outside that scope of the present document.

9.13.1.2 Divergences from OCAP Solution

The GEM design is based as closely as possible on OCAP [5] however some technical changes were found to be necessary to address different market requirements. These are as follows:

- Service bound applications are more important in GEM than in OCAP. One specific concern with unbound applications is their impact on the testing of service bound applications. The GEM design diverges from OCAP in order to minimize the impact on testing of service bound applications.
- GEM unbound applications will be used in IPTV where signalling is XML based rather than MPEG-2 based. The GEM design diverges from OCAP in its use of XML for signalling in an IPTV environment.
- Even though IPTV deployments today are based around a single network operator owning the receiver, the DVB commercial requirements for IPTV include scalability to future situations where services from multiple IPTV operators can be received. The GEM design diverges from OCAP in order to permit this scalability.
- GEM does not support CableCARD or the out of band channel (OOB) hence aspects of OCAP related to or relying on these cannot be re-used directly in GEM.
- Some aspects of OCAP are a direct consequence of specific US regulatory requirements. In particular the absence of support for operator applications built in to the box at the time of manufacture. This constraint would not apply to GEM-IPTV.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

9.13.1.3 Overview

The present document defines a single environment within which there are multiple service contexts. These may include:

- Service contexts presenting broadcast services.
- Service contexts presenting stored application services.
- Service contexts presenting return channel signalled services.
- Service contexts presenting internet client services (in GEM terminals supporting the internet access profile).
- Service contexts presenting recorded services (in GEM terminals supporting the GEM PVR/PDR extension).
- Service contexts presenting abstract services (containing unbound applications).

When the GEM environment is initialised, abstract services are created based on the signalling in the XAIT (see clause 9.13.4, "Initialization of GEM Environment" below). Abstract services may also be created by GEM applications on GEM terminals supporting the privileged application extension.

Where there is a subsidising service provider:

- they can control the single environment using the features of the privileged application option;
- they are responsible for managing the set of unbound applications. Other service providers can only deploy unbound applications with their agreement.

In the absence of control by a subsidising service provider, the controlling role is performed by the GEM implementation as is the case in previous versions of GEM.

9.13.2 Service model

Clause 10.2.2.2 of OCAP [5] "OCAP Service Model" may be supported.

9.13.3 Application lifecycle

Clause 10.2.2.3 OCAP [5] "Application Signaling and Lifecycle" may be supported.

9.13.4 Initialization of GEM Environment

When the GEM environment is initialized, (either from scratch or after a re-start) the following steps shall apply:

- The XAIT shall be loaded and parsed including updating the service list and the applications database.
- The most recent initial monitor application (if any) shall be identified (either signalled in the XAIT and present in the network or already stored in the GEM terminal). If one is identified then it shall be launched and the implementation shall wait for it to signal that it is ready by calling `org.ocap.OcapSystem.monitorConfiguredSignal`, (see clause 11.15.5, "OCAP annex O - the org.ocap package" in the present document).
- For each abstract service in the service list that is signalled as "`auto_select`" a service context shall be created and the abstract service shall be selected in that service context. If the version of a XAIT signalled application in application storage matches that defined in the XAIT, then the stored version may be launched. Otherwise, the signalled version shall be launched.

NOTE: The contents of this clause are aligned with clause 20.2.2 of OCAP [5] however the number of dependencies on CableCARD in that clause make it impractical to reference that text.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 103 of 861

10 Application signalling

10.1 Introduction

This clause covers the following topics:

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

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

10.1.1 Summary of requirements on common signalling

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

- Some form of Application Description (see clause 10.4) with information sufficient to:
  - identify the source of the application code and other assets;
  - identify the application's application ID and 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 summarized as follows:

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

10.2 Program specific information

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

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

10.3 Locators within an Application Description

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

10.4 Application Description

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

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

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

10.4.1 Application Description transmission and monitoring

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

10.4.2 Visibility of Application Description and tuning

If an application tunes away from a transport stream where its signalling is carried without selecting a new service, it shall be permitted to continue running even if the Application Description is no longer available to the GEM terminal.

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

10.4.3 Content of the Application Description

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

Table 11: Application description

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

application_type: Identifies the type of application. Specifications based on GEM shall provide a mechanism for indicating at least two application types: DVB-J and DVB-HTML. For information purposes current registrations for these application types are shown in table 12.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 12: Application types

Application_type	Description
0x0000	Reserved_future_use
0x0001	DVB-J application
0x0002	DVB-HTML application
0x0003 to 0x7FFF	Subject to registration with DVB

These values are formally maintained by the DVB Project [18].

organisation_id: This 32 bit field is a globally unique value identifying the organization that is responsible for the application. These values are registered in TS 101 162 [29]. Values of zero shall not be encoded.

This field is reproduced in the `organisationName` field of the subject name in the "leaf" certificate of an authenticated application (see clause 12.5.6, "subject").

NOTE 1: The inclusion of this field in the leaf certificate provides authentication of the value.

NOTE 2: `organisation_id` values between 0x80000000 and 0xffffffff are known to create problems in some implementations and their use is strongly discouraged.

application_id: This 16 bit field uniquely identifies the application function. This is allocated by the organisation registered with the `organisation_id` who decides the policy for allocation within the organisation. Values of zero shall not be encoded.

The application id values are divided into two ranges: one for unsigned applications and one for signed applications. This is for security reasons (see clause 12.1.1, "Overview of the security framework for applications"). Applications transmitted as unsigned shall use an application id from the unsigned applications range and applications transmitted as signed shall use an application id from the signed applications range.

Table 13: Value ranges for application_id

Application_id values	Use
0x0000	Shall not be used
0x0001 to 0x3fff	Application_ids for unsigned applications
0x4000 to 0x7fff	Application_ids for signed applications
0x8000 to 0xfffd	Reserved for future use by DVB
0x8000 to 0x9fff	Application_ids for privileged applications
0xa000 to 0xfffd	Reserved for future use by DVB
0xfffe	Special wildcard value for signed applications of an organization
0xffff	Special wildcard value for all applications of an organization

Application id values 0xffff and 0xfffe are wild cards. They shall not be used to identify an application but are allowed for use in other descriptors. The value 0xffff matches all applications with the same `organisation_id`. The value 0xfffe matches all signed applications with the same `organisation_id`.

The same application identifier may be used in different application types for applications performing essentially the same function.

application_control_code: A DVB-J application control code, as defined in clause 10.4.3.1.

Support for the `REMOTE` application type is not required, but may optionally be present in GEM terminal specifications.

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

application_profile: This 16 bit field is an integer value which represents the application type specific profile. This indicates that a receiver implementing one of the profiles listed in this loop is capable of executing the application.

version.major: This 8 bit field carries the numeric value of the major sub-field of the profile version number.

version.minor: This 8 bit field carries the numeric value of the minor sub-field of the profile version number.

version.micro: This 8 bit field carries the numeric value of the micro sub-field of the profile version number.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

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

service_bound_flag: If this field is set to "1", the application is associated only with the current service and so the process of killing the application shall start at the beginning of the service change regardless of the contents of the destination Application Description.

visibility: This 2 bit field specifies whether the application is suitable to be offered to the end-user for them to decide if the application should be launched. Table 14 lists the allowed values of this field.

Table 14: Definition of visibility states for applications

Visibility	Description
00	This application shall not be visible either to applications via an application listing API or to users via the navigator with the exception of any error reporting or logging facility, etc.
01	This application shall not be visible to users but shall be visible to applications via an application listing API.
10	Reserved_future_use.
11	This application can be visible to users and shall be visible to applications via the application listing API.

NOTE 3: For example, in a service offering a number of games to the end-user, these values would be used as follows:

00 - the autostart generic launcher application offering the end user the choice of which game to launch.

11 - the games which the end-user can chose between.

01 - the common server application which all the games use to communicate with a server over the interaction channel.

application_priority: This field identifies a relative priority between the applications signalled in this service. Terminal specifications based on GEM shall support at least 32 priority levels.

- Where there is more than one application with the same Application identification this priority shall be used to determine which application is started.
- Where there are insufficient resources to continue running a set of applications, this priority shall be used to determine which applications to stop or pause.
- A larger integer value indicates higher priority.
- If two applications have the same application identification and the same priority, the GEM terminal may make an implementation-dependent choice on which to start.
- When used in an XAIT, the value of 255 shall identify the initial monitor application. The value of 255 has no special meaning for applications signalled in an AIT.

NOTE 4: Requirements on signalling unbound applications with this priority are found in clause 10.2.2.3.1 "Applications Signaled in the XAIT" of OCAP [5] which is referenced by clause 9.13.3, "Application lifecycle".

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 107 of 861

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 clause 10.4.3.2.

application_icon_flags: Flags describing the icon files in the directory identified by the `application_icon_locator`, in the format specified in clause 10.4.3.2.

NOTE 5: GEM terminal specifications may define application icon signalling that also allows different icon sizes to be indicated.

10.4.3.1 DVB-J application control codes

The application control codes for DVB-J applications are listed in table 15.

Table 15: DVB-J application control code values

Code	Identifier	Semantics
0x00		Reserved_future_use.
0x01	AUTOSTART	The file system element(s) (e.g. an Object Carousel module) containing the class implementing the Xlet interface is loaded, The class implementing the Xlet is loaded into the VM and an Xlet object is instantiated, and the application is started subject to usual restrictions, etc.
0x02	PRESENT	Indicates that the application is present in the service, but is not autostarted.
0x03	DESTROY	When the control code changes from AUTOSTART or PRESENT to DESTROY, the destroy method of the Xlet is called (with the **`unconditional`** parameter set to **false**) by the application manager and the application is allowed to destroy itself gracefully.
0x04	KILL	When the control code changes from AUTOSTART or PRESENT to KILL, the destroy method of the Xlet is called (with the **`unconditional`** parameter set to **true**) by the application manager.
0x05		reserved_future_use.
0x06	REMOTE	This identifies a remote application that is only launchable after service selection.
0x07 to 0xFF		Reserved_future_use.
0x07	DISABLED	Application shall not be started and attempts to start it shall fail. Such applications need not be signalled with a dvb_j_application_descriptor and the initial_class_byte in the dvb_j_application_location_descriptor is never used.
0x08 to 0xFF		Reserved_future_use.

See clause 9.2.3, "DVB-J Application Lifecycle".

10.4.3.2 Application icons descriptor

Zero or one instance of this descriptor shall be included in the application information of an application. It allows icons to be associated with the application. The content format for these possible icons shall be restricted PNG as specified in clause 15.1, "PNG - restrictions". Table 16 shows the syntax of the application icons descriptor.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 16: Application icons descriptor syntax

	No.of Bits	Identifier	Value
application_icons_descriptor() {			
descriptor_tag	8	uimsbf	
descriptor_length	8	uimsbf	
icon_locator_length	8	uimsbf	
for (i=0; i<N; i++) {			
icon_locator_byte	8	uimsbf	
}			
icon_flags	16	bslbf	
for (i=0; i<N; i++) {			
reserved_future_use	8	bslbf	
}			
}			

descriptor_tag: This 8 bit integer with value 0x0B identifies this descriptor.

icon_locator_length: This 8 bit integer specifies the number of bytes in the string that prefixes standard icon file name.

icon_locator_byte: This 8 bit value is one byte of the icon locator string.

The icon locator is the first part of the string that specifies the location of the icon files. This is application type dependent. See table 17. The `icon_locator` shall not end with a "/" slash character.

Table 17: Icon locator semantics

Application_type	Description
0x0000	Reserved_future_use.
0x0001	For DVB-J this is a path relative to the base directory of the application as defined in MHP [1] clause 10.9.2, "DVB-J application location descriptor".
0x0002	For DVB-HTML this is a path relative to the physical root of the application as defined in MHP [1] clause 10.10.2, "DVB-HTML application location descriptor".
0x0003 to 0xFFFF	Reserved_future_use.

icon_flags: This 16 bit field carries a value which is the bitwise OR of the flag bits that identify the icons that are provided for the application. The flag bits are defined in table 18.

Table 18: Definition of different icon flags

Icon flag bits	Description of icon size and pixel aspect ratio
0000 0000 0000 0001	32 x 32 for square pixel display
0000 0000 0000 0010	32 x 32 for broadcast pixels on 4:3 display (see note)
0000 0000 0000 0100	24 x 32 for broadcast pixels on 16:9 display
0000 0000 0000 1000	64 x 64 for square pixel display
0000 0000 0001 0000	64 x 64 for broadcast pixels on 4:3 display
0000 0000 0010 0000	48 x 64 for broadcast pixels on 16:9 display
0000 0000 0100 0000	128 x 128 for square pixel display
0000 0000 1000 0000	128 x 128 for broadcast pixels on 4:3 display
0000 0001 0000 0000	96 x 128 for broadcast pixels on 16:9 display
0000 0010 0000 0000	256 x 256 for square pixel display
0000 0100 0000 0000	256 x 256 for broadcast pixels on 4:3 display
0000 1000 0000 0000	192 x 256 for broadcast pixels on 16:9 display
xxxx xxx0 0000 0000	reserved_future_use
xxxx 0000 0000 0000	reserved_future_use
NOTE: Approx. 15/16 pixel aspect ratio on 50 Hz system.	

The file names for the icon files are encoded in a standard way:

```
filename    = icon_locator "/dvb.icon." hex_string
hex_string  = 4*4hex
hex         = digit | "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f"
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 109 of 861

```
digit       = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

An icon file shall contain exactly one icon. The icon contained in the icon file shall have the format specified by the 4 hexadecimal digit postscript of its file name.

NOTE: This means that if the icon_flags field of the application_icons_descriptor were to have a value indicating the presence of multiple icons, each of the indicated icons would have its own icon file. For example, if icon_flags has a value of 0x0005, the directory specified by `icon_locator` would contain two files named dvb.icon.0004 (for 24 x 32 square pixel rendering) and dvb.icon.0001 (for 32 x 32 square pixel rendering).

10.4.3.3 Graphics constraints descriptor

The graphics constraints descriptor defines the circumstances under which an application can work (or has been tested to work). These circumstances are:

- which full screen graphics resolutions an application supports;
- whether an application can work when its video is controlled (e.g. scaled to less than full screen size) by another application running external to the first application's service context (e.g. EPG or navigator).

Table 19: Graphics constraints descriptor syntax

	No. of Bits	Identifier	Value
graphics_constraints_descriptor() {			
descriptor_tag	8	uimsbf	0x14
descriptor_length	8	uimsbf	
reserved_future_use	5	bslbf	
can_run_without_visible_ui	1	bslbf	
handles_configuration_changed	1	bslbf	
handles_externally_controlled_video	1	bslbf	
For(i=0;i<N;i++) {			
graphics_configuration_byte	8	Uimsbf	
}			
}			

Where the fields have the following meanings:

descriptor_tag: This 8 bit integer with value 0x14 identifies this descriptor.

descriptor_length: This 8 bit field indicates the number of bytes following the descriptor length field.

can_run_without_visible_ui: If this 1 bit field is set to 1 then the application can usefully run with no user interface visible. If this field is set to 0 then the application is not useful with no user interface visible. Applications signalled with this bit set are responsible for registering listeners to detect when it would be reasonable to show their user interface again.

handles_configuration_changed: If this 1 bit field is set to 1 then the application can handle changes in the graphics configuration between those signalled in this descriptor. If set to zero then once the default graphics configuration has been set for an application instance, it will only correctly display under that graphics configuration.

handles_externally_controlled_video: If this 1 bit field is set to 1 then the application can handle being displayed when the presentation of the video in the same service is controlled by an application external to that service. Examples include picture in picture and picture outside picture.

graphics_configuration_byte: This 8 bit field contains a value from the following table. The full screen configurations are sorted from most preferred to least preferred.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 20: graphics configuration byte values

Value	Meaning
0	Reserved
1	Full screen standard definition
2	Full screen 960x540
3	Full screen 1280x720
4	Full screen 1920x1080
5 to 31	Reserved for future use by DVB project
32 to 255	Reserved for future use

Applications where this descriptor is not signalled shall be assumed to have a descriptor containing one `graphics_configuration_byte` whose value is 1, `can_run_without_visible_ui` being '0', `handles_configuration_changed` being 0 and `handles_externally_controlled_video` being 0.

Applications where the loop of graphics configuration bytes is empty shall be assumed to not care about a default graphics configuration. Either they do not use graphics or they are written to support the full range of graphics configurations defined in the present document and tested accordingly.

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.

NOTE: To efficiently support this feature on services that do not contain the application code, it may be desirable to have signalling equivalent to that described in MHP [1], clause 10.7.5.

10.4.5 AIT File

10.4.9.1 Syntax

The interaction channel encoding of the Application Description into the AIT File is as follows:

- A single file shall contain all of the data.
- The file shall contain a concatentation of Application Information Sections (specified in MHP [1], clause 10.4.6, "Syntax of the AIT").
- The possibly multiple sections shall be ordered as follows:
  - Ascending order of `application_type`.
  - Within a single value of `application_type` in ascending order of `section_number`.
- All sections shall have `current_next_indicator` set to '1'.

10.4.9.2 Syntactic restrictions

10.4.9.2.1 Transport protocols

The only allowed transport protocol type has id value 0x0003. The protocol_id is an identifier of the protocol used for carrying applications. The values of the `protocol_id` are shown in table 21.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 21: Protocol_id

Protocol_id	Description
0x0000	Reserved_future_use.
0x0001 to 0x0002	Reserved for use by DVB (see note).
0x0003	Transport via HTTP over the interaction channel as described in clause 10.8.1.3, "Transport via interaction channel".
0x0004 to 0x00FF	Reserved for use by DVB.
0x0100 to 0xFFFF	Subject to registration in TS 101 162 [29].
NOTE: MHP [1] uses the protocol_id values 0x00001 for the MHP Object Carousel and 0x0002 for IP via DVB multiprotocol encapsulation.	

10.4.9.3 Semantics

The AIT File shall be loaded once during service selection. There is no requirement to monitor or poll the AIT File for any subsequent changes except as part of a subsequent service selection operation referring to the same AIT File. Consequences of this are:

- Only the AUTOSTART and PRESENT application control codes (of those defined in clause 10.4.3.1) are appropriate.
- No standard mechanism for dynamic lifecycle control is provided.

NOTE: If dynamic lifecycle control is required application private mechanisms could be used. For example, a TCP connection over the interaction channel to a controlling server.

- Changes made to the AIT File after service selection shall not be detected or reported.

10.4.9.4 MIME type

The MIME type for an AIT File shall be "application/vnd.dvb.ait". The file extension shall be ".ait".

Earlier versions of GEM and derived specifications defined the MIME type for AIT file as application/dvb.ait without the vnd. prefix. This type name is deprecated and its use is strongly discouraged. If a GEM terminal gets a deprecated MIME type, it shall behave as if the vnd. prefix was present.

10.5 DVB-J specific Application Description

10.5.1 General

Additional signalling specific to DVB-J applications shall be present in GEM terminal specifications.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

10.5.2 Content of DVB-J Application Description

For each Application Description that refers to a DVB-J application, it shall be possible to signal information sufficient to derive the following.

Table 22: DVB-J Application Description

Function	Type
for (i=0; i<N; i++) {	
dvbj_app_parameter	String
}	
base_directory	Locator
for (i=0; i<N; i++) {	
classpath_element (optional)	Locator
}	
initial_class_name	String
can_run_without_visible_ui	Boolean
handles_configuration_changed	Boolean
handles_externally_controlled_video	Boolean
For(i=0; i<N; i++) {	
graphics_configuration_byte	Byte
}	

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.

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

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

classpath_element: GEM-based terminal specification may include optional signalling to indicate a list of other locators to be added to an application's class path.

EXAMPLE: MHP [1], clause 10.9.2 defines the `classpath_extension` for this purpose.

If support for this is included in a terminal specification, there may be restrictions placed on these locators, e.g. that they represent sub-directories of the `base_directory`.

can_run_without_visible_ui: If this value is true then the the application can usefully run with no user interface visible. If this value is false then the application is not useful with no user interface visible. Applications signalled with this value set to true are responsible for registering listeners to detect when it would be reasonable to show their user interface again.

handles_configuration_changed: If this value is true then the application can handle changes in the graphics configuration between those signalled in this descriptor. If false then once the default graphics configuration has been set for an application instance, it will only correctly display under that graphics configuration.

handles_externally_controlled_video: If this value is true then the application can handle being displayed when the presentation of the video in the same service is controlled by an application external to that service. Examples include picture in picture and picture outside picture.

graphics_configuration_byte: This 8 bit field contains a value from the following table. The full screen configurations are sorted from most preferred to least preferred.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 23: Graphics configuration byte values

Value	Meaning
0	Reserved
1	Full screen standard definition
2	Full screen 960x540
3	Full screen 1280x720
4	Full screen 1920x1080
5 to 31	Reserved for future use by DVB project
32 to 55	Reserved for future use

10.6 Constant Values

GEM terminal specifications that include the functional equivalent named "Application Signalling" as defined in clause 15.6, "Functional equivalents" and that define additional descriptors in the Application Description shall register the descriptor tag values with the DVB. For information purposes current registrations are shown in table 24.

Table 24: Registered AIT Descriptor Tags

AIT Descriptor Tag Values	Organization
0x60 to 0x65	ATSC
0x66 to 0x6D	CableLabs
0x6E to 0x7F	reserved

These values are formally maintained in TS 101 162 [29].

10.7 Plug-in signalling

Two signalling scenarios are defined for delegated applications and plug-ins:

- Native signalling scenario.
- GEM signalling scenario.

GEM terminal specifications may substitute functionally equivalent signalling, if required.

10.7.1 Native signalling scenario

In this scenario it is sufficient for the plug-in to be signalled as a normal GEM application. Optionally it could also be signalled as a plug-in using the Plug-in descriptor (see clause 10.7.4).

10.7.2 GEM signalling scenario

In this scenario GEM plug-in signalling can replace some or all of the native signalling of the delegated application. The GEM signalling allows the GEM terminal to:

- Identify that one (or more) delegated applications are available.
- The plug-in that is required.
- How to start the plug-in.
- How to introduce the delegated application to the plug-in.

The plug-in may use native signalling defined for the delegated application to locate, load and execute it. This is outside the scope of the present document.

Each delegated application is associated with an Application Description (clause 10.4).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The additional GEM signalling provided to support plug-ins is:

- "Delegated application descriptor" (see clause 10.7.3).

  An optional descriptor for delegated applications.

- "Plug-in descriptor" (see clause 10.7.4).

  A mandatory descriptor for plug-ins using GEM signalling.

10.7.3 Delegated application descriptor

Zero or one instance of this descriptor can be placed in the Application Description of the delegated application.

This optional descriptor allows one or more specific plug-ins to be identified for this delegated application. This overrides the default identification of the plug-in based on the application type signalled in the Application Description for the delegated application. If this descriptor is absent then no specific plug-in is identified.

Where more than one application is identified they are listed in order from most preferred to least preferred. If none of the preferred applications is available to the GEM terminal then it shall use its default mechanism to identify the plug-in for this delegated application.

Table 25: Plug-in reference descriptor

	No.of Bits	Identifier
delegated_application_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0; i<N; i++){		
application_type	48	uimsbf
}		
}		

descriptor_tag: This 8 bit integer with value 0x0E identifies this descriptor.

descriptor_length: This 8 bit field indicates the number of bytes following the descriptor length field.

application_identifier: This 48 bit field identifies a specific application that can consume this application.

10.7.4 Plug-in descriptor

One or more plug-in application descriptors shall be placed in the Application Description of each plug-in.

This descriptor defines, for an application that implements a plug-in, the set of delegated application formats that it supports. The descriptor identifies both the type of delegated application supported and the set of profiles and versions supported. The normal rules for matching an application to a compatible platform apply when matching a delegated application with a suitable plug-in.

DVB-J applications signalled with this descriptor shall implement the API defined in clause 11.7.8, "Plug-in APIs".

The plug-in shall be considered suitable for running content if the `application_type` matches, and the version numbers are compatible.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 26: Plug-in application descriptor

	No.of Bits	Identifier
plugin_application_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
application_type	16	uimsbf
for(i=0; i<N; i++){		
application_profile	16	uimsbf
version.major	8	uimsbf
version.minor	8	uimsbf
version.micro	8	uimsbf
}		
}		
}		

descriptor_tag: This 8 bit integer with value 0x0F identifies this descriptor.

descriptor_length: This 8 bit field indicates the number of bytes following the descriptor length field.

application_type: This 16 bit field identifies a delegated application type that this application can consume.

application_profile: This 16 bit field is an integer value which represents the application type specific profile.

version.major: This 8 bit field carries the numeric value of the major sub-field of the profile version number.

version.minor: This 8 bit field carries the numeric value of the minor sub-field of the profile version number.

version.micro: This 8 bit field carries the numeric value of the micro sub-field of the profile version number.

10.8 Stored Applications

GEM terminal specifications may substitute functionally equivalent signalling, if required.

10.8.1 Use of stored application signalling

A GEM terminal that doesn't provide storage or doesn't recognize these extensions will perceive storable applications as viable applications that potentially can be run from the service-bound connection.

10.8.1.1 Stored broadcast service related applications

For stored service-bound applications the service-bound Application Description information shall be used when launching the stored application. So, little information from the Application Description needs to be stored with the application.

10.8.1.2 Stored stand-alone applications

For stored stand-alone applications, the Application Description information used when launching the application shall come from a stored representation of the Application Description. While a stored service is being presented in a service context, the following apply:

- The applications database shall be populated with information from this stored representation of the Application Description for the applications forming part of that stored service.
- The implementation shall report any changes in the stored representation of the Application Description in the same way as it would report changes in a service-bound Application Description.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

10.8.2 Application storage descriptor

The application storage descriptor advertises that an application can be stored and provides some indications of its properties. The presence of this application storage descriptor indicates that an Application Description File is provided for the application (see clause 10.8.3). For a storable application a single application storage descriptor shall be placed in the Application Description.

This descriptor, and the implied Application Description File, also supports receivers that implement speculative caching.

Table 27: Syntax of application storage descriptor

	No.of Bits	Identifier
application_storage_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
storage_property	8	uimsbf
not_launchable_from_broadcast	1	bslbf
launchable_completely_from_cache	1	bslbf
is_launchable_with_older_version	1	bslbf
reserved	5	bslbf
reserved	1	bslbf
version	31	uimsbf
priority	8	uimsbf
}		

descriptor_tag: This 8 bit integer with value 0x10 identifies this descriptor.

descriptor_length: This 8 bit field indicates the number of bytes following the descriptor length field.

storage_property: This 8 bit field indicates the characteristics of the application w.r.t storage.

Table 28: Semantics of storage property values

Storage_property	Property
0	broadcast related
1	stand alone
2 to 255	reserved

If the storage property is "broadcast related" then the application's life cycle is controlled by the broadcast service. Such applications are suitable for caching but not for running as a stand alone application.

If the storage property is "stand alone" then the application can usefully be launched by the user independently of a broadcast service. Applications with this property may also be launched as if broadcast related if the currently selected service lists them in its Application Description.

launchable_completely_from_cache: This 1 bit field is a flag. When set to "1", this indicates that this application can be run entirely from cache, without connecting to the transport protocol signalled in the application's Application Description, if all the critical files have been cached. If set to "0", a connection to this transport protocol must be made in order to run this application as a broadcast-related application. This flag only applies if running the application as a broadcast-related application, it is ignored when storing an application into a stored service, as all applications signalled as "stand alone" can run without a connection to the transport protocol when run as part of a stored service.

This flag shall only be set to "1" when `not_launchable_from_broadcast` is also set to "1".

NOTE: This flag should be set to "1" only for applications where the object carousel is not present at all.

is_launchable_with_older_version: This 1-bit field is a flag. When set to "1", the GEM terminal shall start the cached application where the version number of the cached application is lower than the version number of the broadcast application. If the version number of the cached application is higher than the version number of the broadcast application, the cached application shall not be started.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 117 of 861

If set to "0", the cached application shall not be started. Informative: If this flag is set, the cached application is responsible to handle version conflicts between cached application code and broadcast application data.

not_launchable_from_broadcast: This 1 bit field is a flag. When set to "1" this indicates that the delivery characteristics of this application are such that it is not useful to launch the application unless it has been completely cached/stored. If set to '0' then caching provides some benefit but is not essential.

Applications in stored services and applications where the not_launchable_from_broadcast bit is set to "1" shall only be launched where the GEM terminal has stored the complete set of files which are listed in the Application Description File as being critical.

version: This 31 bit field provides the version number of the application. This number starts at zero and increments by one each time any of the files listed in the Application Description File change or the contents of the Application Description File itself changes. Used values shall never be reused, in the event that the number range is exhausted a new application id shall be used.

priority: This field indicates the priority of this application for storage relative to the other applications signalled in this service.

It is only meaningful for applications which have been proactively cached by the GEM terminal implementation and shall be ignored otherwise.

Higher values indicate more important applications to store. The behaviour when applications have the same priority is implementation dependent.

Table 29: Storage descriptor flag combinations

Not_launchable_from_broadcast	Launchable_completely_from_cache	Is_launchable with_older_version	Description
0	0	0	Normal case.
0	0	1	Shall not be signalled.
0	1	0	Shall not be signalled.
0	1	1	Shall not be signalled.
1	0	0	Runs if signalled version is stored.
1	0	1	Runs if signalled or older version is stored.
1	1	0	Runs completely from cache if signalled version is stored. The application cannot be stored due to unavailability of DSM-CC for the current service.
1	1	1	Runs if signalled or older version is stored. The application cannot be stored due to unavailability of DSM-CC for the current service.
When set, flag indicates that files are present but bitrate is too low.	When set, flag indicates that files are not present in current broadcast at all.		

10.8.3 Application Description File

10.8.3.1 Description

The Application Description File provides the list of files that need to be installed as well as other related necessary information. The notation uses an XML-based syntax.

For those applications that can be stored, an Application Description File file shall be placed in the same carousel as the application.

NOTE: The application description file does not duplicate all the information needed to run the application, the GEM terminal needs to use also the information in the Application Description when installing the application.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Where a file is listed in the Application Description File of more than one application and is stored, the GEM terminal shall ensure that each application sees the correct version of the file for that application. The version of the file visible to one application shall not be changed by any changes in the version of the file visible to any other application which may share that same file.

10.8.3.2 Application Description File name and location

The application description file shall be located in the base directory of a DVB-J application (as signalled in the dvb-j application location descriptor). By convention, the name of an ADF is:

```
'dvb.storage.oooooooo.aaaa'
```

where:

```
oooooooo is the organisation id of the application as a 8 character hexadecimal string
aaaa is the application id as a 4 character hexadecimal string
```

The organisation id and application id shall be padded with leading zeros to the specified length.

Lowercase hex digits are used to encode the organisation id and application id.

See also clause 10.8.2, "Application storage descriptor".

10.8.3.3 Syntax

The syntax of the Application Description File is defined by the following XML DTD.

The `PublicLiteral` to be used for specifying this DTD in document type declarations of the XML files is:

```
"-//DVB//DTD Application Description File 1.0//EN"
```

and the URL for the SystemLiteral is:

```
"http://www.dvb.org/mhp/dtd/applicationdescriptionfile-1-0.dtd"
```

```
<!ENTITY % object "(dir|file)">

<!-- the main element for the application description -->
<!ELEMENT applicationdescription (%object;)+>

<!ATTLIST applicationdescription version NMTOKEN #REQUIRED>

<!ELEMENT dir (%object;)*>
<!ATTLIST dir
    name CDATA #REQUIRED
    priority NMTOKEN #IMPLIED
>

<!ELEMENT file EMPTY>
<!ATTLIST file
    name CDATA #REQUIRED
    priority NMTOKEN #IMPLIED
    size NMTOKEN #REQUIRED
>
```

10.8.3.4 Semantics

version: A decimal integer denoting the version number of this application.

Must not contain leading zeros (unless it is "0"). Must match the version number signalled in the version field of the application storage descriptor in the Application Description entry of this application otherwise the application description file is invalid. (This field allows application authors to ensure that the version number signalled in the Application Description is correct. If it is wrong then this prevents any files being stored.)

name: This attribute provides the name of a file system object (directory or file) that is storable. This is the name of the object within its enclosing directory and hence does not include any directory path information. For the name attribute of a file element only, the last character of the name can be the wild-card character "*". This character will match any string including an empty string.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

If name is "." or "..", contains the path separator character "/", or contains the character NUL (U+0000), then receivers shall reject the ADF as invalid.

NOTE 1: No elements are provided for naming object types such as Stream or StreamEvent therefore there is no mechanism to specify that Stream and StreamEvent objects are required to be stored.

NOTE 2: Listing a directory object in the file does not imply anything about those contents of the directory which are not themselves listed in the file.

For requirements on filenames of stored applications, see clause 14.6, "Filename requirements".

priority: This attribute describes how important it is to store this object. The value must be between 0 and 255, inclusive. If it is outside this range then the application description file is invalid. The value zero indicates that it is critical to store the object (i.e. there is no benefit in storing any objects unless this part is stored). Higher values indicate lower storage priority.

The default value for the priority attribute is zero (i.e. critical).

The priority for an object inherits from the immediately enclosing directory.

size: This attribute defines the size in bytes of the file, or files where the name attribute includes a wild-card.

Paths are relative to the directory containing the application description file, i.e. <file> and <dir> elements immediately inside the <applicationdescription> element refer to files and directories in the same directory as the application description file.

10.9 Signalling for providers

If a GEM terminal includes support for providers, appropriate signaling for a provider export descriptor and a provider usage descriptor must be defined in the GEM terminal specification.

The provider export descriptor is used in the Application Description for an application which exports (contains) an XletBoundProvider. It declares that the classes of the XletBoundProvider are found in that application.

A provider export descriptor carries information to derive the following information:

provider_name: A string specifying the name of the provider.

provider_class_name: A string specifying the fully qualified name of the class in the application that implements the named provider.

The provider usage descriptor is included in the Application Description for an application which uses an XletBoundProvider and does not include the classes of the provider as part of itself.

A provider export descriptor carries information sufficient to derive the following information:

provider_name: A string specifying the name of the provider.

10.10 Signalling for IPTV

10.10.1 Service bound application signalling

If the GEM terminal specification adopts SD&S signaling (see TS 102 034 [80]),

service bound applications shall be signalled by including either an ApplicationList or an ExternalapplicationAuthorisedList in either the IPService or the Package elements of SD&S (see TS 102 034 [80]). This is fully specified in annex AR "(normative): XML encoding for AIT" of the present document.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

10.10.2 XAIT

If the GEM terminal specification adopts SD&S signaling (see TS 102 034 [80]), unbound applications shall be signalled by including an ApplicationList in the SD&S service provider discovery record for the subsidising service provider. This is fully specified in annex AR, "XML encoding for AIT". Any ApplicationList elements in the service provider discovery records for other service providers shall be ignored.

Monitoring for changes in the XAIT shall be performed as defined in clause 5.4.3 of TS 102 034 [80].

NOTE: See org.ocap.application.AppSignalHander for notification of XAIT updates to applications.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11 DVB-J platform

11.1 The virtual machine

The Java platform is defined in PBP 1.1 [4].

11.2 General issues

11.2.1 Basic Considerations

Unless otherwise specified, GEM implementations are not required to include any classes or methods marked as deprecated in those API specifications which are either referenced by or included in the present document. Where a class is defined by the present document as implementing a specific interface, and that interface requires the class to provide an otherwise deprecated method, the interface overrides the deprecated mark and the method is required by the present document.

Each DVB-J application instance is considered to logically run in its own virtual machine instance. For this reason, it cannot rely on finalizers that are defined in application classes being run when the application terminates. When the application manager terminates the entity that represents the virtual machine in which the application is run, a conformant implementation is permitted to not run application finalizers, as spelled out in section 2.17.9 of the Java virtual machine specification, second edition [10].

DVB-J applications shall not synchronize on system classes or other exposed system static objects else undefined behaviour may occur.

`SecurityExceptions` shall only be thrown either where they are specified by the defining document or where identified in the present document for selected references which do not include any `SecurityExceptions`.

Unless specified otherwise for a referenced specification, it is an allowable implementation choice to not override methods as long as the defined semantics are respected. The implication of this is that such differences between implementations will be visible when using the `java.lang.reflection` package.

The inclusion of java packages shall not imply the inclusion of other subpackages, e.g. the inclusion of the `javax.tv.media` package does not automatically imply the inclusion of the `javax.tv.media.protocol` subpackage. Inclusion of subpackages shall be explicitly listed in the present document.

DVB-J applications shall not use additional public or protected methods or fields in the `org.dvb` namespace which are not listed in the present document. Applications which scale to run on multiple revisions of the present document shall only use `org.dvb` methods etc. appropriate to the version of the platform on which they are running.

Applications shall not define classes or interfaces in any package namespace defined in the present document. GEM terminals shall enforce this using the `SecurityManager.checkPackageDefinition` mechanism.

NOTE: DVB-J applications that are written to be scalable across multiple GEM profiles and/or versions of profiles and/or optional features may include references to API classes that are only present if the GEM terminal implements the profile or version of a profile or an optional feature that these API classes represent. Such applications have to be written with care using a standard Java idiom for this case. This will ensure that they work on all MHP and GEM terminal implementations, including those that use "eager linking" technology. A simple idiom to achieve this is presented in clause W.3, "Example of testing for optional APIs".

11.2.2 Approach to Subsetting

Where a class included in the present document has methods, fields or constructors with signature dependencies on classes not included in the implemented profile, these methods are not required to be present in an implementation. A compliant implementation choice may require these methods and classes to be present.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Where the present document subsets a package, inclusion of the complete package is allowed but clearly not required. The behaviour of the additional features is not specified for broadcast applications.

11.2.3 Class Loading

11.2.3.1 Fundamental principles

The DVB-J application environment shall be written such that each application appears to run within its own classloader or classloader hierarchy for all classes that are not a part of the platform. As a consequence, two applications will never be able to access the same copy of any application-defined static variable.

In a signed application, all classes or files to be loaded through the classpath shall be signed by at least the set of certificates used to sign the initial xlet class of the application. This applies, for example, to class files comprising the application, and images and other data loaded via the `java.lang.Class.getResource()` mechanism. When authenticating a signed application, a GEM terminal can select any one of these certificates to use to authenticate all subsequent classes or files loaded. The mechanism used to make this selection is implementation dependent.

Where a jar file is included in the search path (e.g. for a `DVBClassLoader`) and is signed according to the GEM security model, all classes or files to be loaded from that jar file shall be considered to be signed by the certificates signing the jar file under the GEM security model.

NOTE 1: GEM imposes no requirements on the use of the signing information within a jar file.

NOTE 2: Where a GEM terminal trusts more than one of the certificates used to sign the initial xlet, it should attempt to select the most trusted of these.

See clause 12, "Security".

Inappropriate, meaningless or illegal locators are silently skipped and searching continues.

11.2.3.2 Class loading and providers

For GEM terminal specifications that include the optional provider APIs (see clause 11.7.9), the above requirement for the classes to be signed with a specific certificate is modified as follows: The classes of the provider shall be signed by a certificate where the `organisation_id` in the subject field of the certificate matches the `organisation_id` in the provider name.

11.2.4 Unloading

NOTE: Class unloading as required by PBP 1.1 [4] is required.

11.2.5 Event listeners

In `org.dvb` and `org.davic` all methods to remove event listeners shall have no effect if the listener is not registered.

NOTE: The number of threads used to send events to event listeners is intentionally implementation dependent. Applications should not block in event listeners as this may prevent other events being delivered.

11.2.6 Event model in DAVIC APIs

Events defined in DAVIC APIs DAVIC 1.4.1p9 [17] which currently directly extend `java.lang.Object` shall extend `java.util.EventObject`. Event listeners defined in DAVIC 1.4.1p9 [17] shall extend `java.util.EventListener`.

11.2.7 Event model in DAVIC and DVB APIs

Each class in `org.dvb` and `org.davic` inheriting from `java.util.EventObject` is just a container for these fields and no validity checks are done for the parameters by this constructor. Instances of these classes are intended to be constructed by the platform implementation and not by applications. The platform implementation will only construct these events with the appropriate information passed in.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

In `org.dvb` and `org.davic`, unless explicitly specified otherwise, all methods to add event listeners add each listener only once if the add method is called with the same parameters multiple times. This means that the same event is delivered only once to each listener even if it has been added twice.

11.2.8 Tuning as a side-effect

This clause applies only to GEM terminal specifications which include a tuner.

No GEM API shall cause tuning unless explicitly specified as such. For example, if a locator that requires tuning is received by the `DVBClassLoader` it shall behave as if the specified file is not available.

11.2.9 Intra application media resource management

Where an application makes conflicting requests for limited media decoding resources, the media decoding resources that are requested most recently are presumed to be the ones that are most wanted. This applies between MPEG-2 I-Frames, MPEG-2 Video "drips" and streaming video. Similarly, this applies between streaming audio and audio from memory.

When a non-broadcast media presentation (audio or video from memory or still image) is interrupted by a resource loss within the same application, the first presentation is cancelled and will not be restored.

If a broadcast media presentation is interrupted by a resource loss within the same application, the broadcast presentation is restored when the interrupting presentation ends.

11.2.10 Application thread priority

The `ThreadGroup` of application threads shall have a `MaxPriority` of `java.lang.Thread.NORM_PRIORITY`.

NOTE: As a consequence applications will not be able to create threads at priorities greater than `java.lang.Thread.NORM_PRIORITY` since they don't have `java.lang.RuntimePermission( "modifyThread" )`. Applications may perform compute intensive tasks within application created threads without being considered unresponsive.

11.2.11 Text Encodings

Where the specification of the Java APIs refers to the default character encoding of the platform, the default for GEM shall be "UTF8" as defined in clause 7.1.5, "Monomedia format for text". The encoding "latin1" shall also be supported as defined in ISO 8859-1 [23].

When present in a Java `String`, the mark-up codes defined in table 80, "Codes defined for use in marked-up text files" shall be encoded in Java chars whose most significant byte is zero and whose least significant byte is the value from table 80. The encoding `"DVBMarkupUTF8"` shall be supported and is defined to be the same as UTF8 except as follows:

- In byte to char translations, the mark-up sequences in table 80 shall be translated into chars as defined above.
- In char to byte translations, sequences of characters matching the encodings above shall be translated into the corresponding mark-up code sequences in table 80.

11.2.11.1 Text encoding in Service Information

This clause applies only for GEM terminal specifications that include the functional equivalent named "SI" as defined in clause 15.6, "Functional equivalents".

Where methods of the DVB SI API or Java TV APIs access strings encoded in the SI tables and return them to applications as String objects, the following character encodings shall be supported as defined in annex A of the DVB SI specification EN 300 468 [14]:

- ISO 6937 [22] (default).
- ISO 8859 [23] through ISO 8859 [23] (SI string first byte codes 0x01...0x05).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- ISO 8859-1 [23] through ISO 8859-9 [23] (SI string first byte code 0x10).
- 16-bit ISO/IEC 10646-1 [15] UCS-2 (SI string first byte code 0x11).

These encodings shall also be supported by the methods in the `org.dvb.si.SIUtil` class. Support of the other character encodings whose signalling is specified in the DVB SI specification is not required from GEM terminals.

11.3 Fundamental DVB-J APIs

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

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

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

11.3.1 Java platform APIs

NOTE: The following packages are defined in PBP 1.1 [4].

11.3.1.1 java.lang package

The `java.lang` package is supported with the following notes and requirements.

a) The following fields shall not be used by inter-operable applications:

- `System.in`.

b) Applications shall be able to use:

- `System.out`;
- `System.err`;
- `Runtime.traceInstructions()`;
- `Runtime.traceMethodCalls()`;

for debugging without any adverse effects to the application. The output shall not be visible to normal end users and shall not conflict with any other API.

c) The `java.lang.Compiler` class and following methods shall be taken as hints from an application to the system. However, as specified by PBP 1.1 [4] there is no guarantee of what happens:

- `Runtime.gc()`;
- `System.gc()`.

d) In addition to the system properties required by PBP 1.1 [4], the following properties are required to be supported for `System.getProperty()`:

- `dvb.returnchannel.timeout` (see clause 11.5.3, "Support for IP over the Return Channel");
- `dvb.persistent.root` (see clause 11.5.6, "Persistent Storage API").
- `dvb.display.aspect_ratio` (see clause 11.4.1.2, "TV user interface").

With the exception of `dvb.persistent.root` all of these properties are accessible to all applications.

`dvb.persistent.root` is accessible only as defined in clause 11.10.2.1, "java.util.PropertyPermission".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

NOTE 1: All GEM terminal specifications require support for the system property "dvb.persistent.root".

Property names beginning "dvb." are reserved for future use.

e) The methods java.lang.SecurityManager.checkPackageDefinition and java.lang.SecurityManager.checkPackageAccess are documented to check a package name against a list of restricted packages.

GEM terminal implementors are recommended to use this mechanism to prevent application classes from having access to implementation classes where this could compromise the security of the GEM terminal.

GEM terminals shall not define classes in the empty ("") package name-space.

NOTE 2: GEM terminal implementors who are also implementing GEM applications need to ensure that the namespaces used for applications do not collide with those used for their GEM implementation.

The call System.currentTimeMillis() shall feature a granularity of the returned time value of not more than 10 ms. The terminal should attempt to keep an accurate clock, e.g. by initialising its clock from a TDT. The default TimeZone for java.util.Calendar shall either be as defined by the end-user or as defined in a TOT if the end-user has not defined any TimeZone. With the call Object.wait(long timeout), the timeout value is specified as a maximum time to wait. The present document further guarantees that if not notified the object will wait for at least timeout 10 ms.

11.3.1.2 java.void

As specified in PBP 1.1 [4], the constants in the java.util.Locale class do not imply support (or otherwise) for these Locales. Locales supported in GEM are specified in profiles (see clause 15.4, "Locale support").

Inter-operable applications shall not call the java.util.TimeZone.setDefault method. The behaviour if this method is called is implementation dependent.

11.3.1.3 Void

11.3.1.4 java.io, javax.microedition.io

The deprecated method java.io.ObjectInputStream.readLine() shall not be called by inter-operable applications.

The classes and interfaces in this package relating to files and file systems have additional semantics defined for GEM specific file systems as follows.

- For broadcast carousels in clause 11.5.1.1, "Constraints on the java.io.File methods for broadcast carousels".
- For persistent storage in clause 11.5.6, "Persistent Storage API".
- For applications running as part of stored services in clause 11.12.2.2, "Modified behaviour of GEM 1.0 APIs".

NOTE: PBP 1.1 [4] requires that javax.microedition.io support a number of protocols, such as "socket:", "http:" and "file:".

11.3.1.5 java.net

Consistent with PBP 1.1 [4], support is required for at least one implementation dependent java.net.URL protocol. Platform methods that return a URL to access resources, such as Class.getResource, ClassLoader.getSystemResource and ClassLoader.getResource shall return instances of java.net.URL using such a protocol where no appropriate protocol is defined in the present document.

In a signed application, a URL to this resource shall be returned as for an unsigned application, regardless of whether the underlying file is signed. When that file is accessed (e.g. via an input stream obtained from java.net.URL.openStream()), then if the file fails authentication as described in clause 11.2.3, "Class Loading", the system shall behave as if the file contained no data (i.e. as if it were a zero-length file).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

NOTE: Due to the overhead of processing the signature verification, asset files which are not critical to be authenticated should be loaded using `java.io.File` or `org.dvb.dsmcc.DSMCCObject` and sent as unsigned.

Support is required for the "`file:`" protocol. Applications shall be able to construct instances of `java.net.URL` using strings containing "`file:`" URLs as defined in RFC 1738 [61] where the `<host>` element is the empty string and the <path> element is an absolute filename.

The return types of `URL.getContent()` are defined by the mappings from data type to java class name listed in table 30.

Table 30: Return types of URL.getContent()

Data type	Return type
Unknown or unsupported data types	java.io.InputStream
Text/plain	java.io.InputStream
Text/vnd.dvb.utf8	java.io.InputStream
Text/Generic	java.io.InputStream
Image/png	java.awt.image.ImageProducer
Image/jpeg	java.awt.image.ImageProducer
Image/mpeg	org.havi.ui.HBackgroundImage

The behaviour of `URL.getContent()` responds to data type using information with the following priority:

a) Content type signalling such as:

- The content type descriptor in the OC (see clause B.2.3, "Content type descriptor").
- The HTTP header (if supported in the profile) the Content-Type header field (if present).

b) The filename extension (if known) (see table 7, "File type identification").

c) Open the file and study.

For GEM terminal specifications that do not include a functional equivalent named "Carousel" as defined in clause 15.6, "Functional equivalents" this requirement regarding the behaviour of `URL.getContent()` does not apply; however, if the functionally equivalent signalling contains data type information, it is recommended that it be given the same priority as the content type descriptor is given.

`getContentType` returns the value contained in the optional content type descriptor in the OC (see clause B.2.3, "Content type descriptor") if present.

`getFileNameMap` returns information derived from table 7 "File type identification".

See the Object Carousel signalling described in clause B.2.3, "Content type descriptor".

11.3.2 GEM platform APIs

11.3.2.1 org.dvb.lang

The `org.dvb.lang` package is supported as defined in annex I, "DVB-J fundamental classes".

Where no parent is specified at creation, the delegation parent classloader of `DVBClassLoader` shall be the original classloader of the calling application.

Classloader delegation is defined in the specification for `java.lang.ClassLoader` in PBP 1.1 [4].

11.3.2.2 org.dvb.event

11.3.2.2.1 Generic description

The `org.dvb.event` package is supported as defined in annex J, "DVB-J event API".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 127 of 861

When sending events from a `org.dvb.event.EventManager` to listeners, the implementation shall ensure that any single listener shall only be sent one platform generated event instance at one time. If a new event is generated before the listener method has returned from processing a previous event, the GEM terminal shall not call that listener method until the call for processing the previous event returns.

The behaviour where applications have multiple such event listeners registered is implementation dependent. Implementations may use multiple threads to send the same event instance to any such multiple listeners. Implementations may ensure that at most one event listener in any one application for all these events is executing at any one time.

The return value of `UserEvent.getWhen()` shall be the difference, measured in milliseconds, between the time when the event occurred and midnight, January 1, 1970 UTC, i.e. like `System.getCurrentTimeMillis()`.

11.3.2.2.2 Additional semantics for org.dvb.event

Privileged applications (and no others) shall be able to exclusively reserve the user input events defined in table 87 "Additional input events required by privileged applications". Attempts by applications without this permission to reserve these user input events shall fail. These events shall not be delivered via the conventional focus mechanism.

11.3.3 Java TV

The Java TV APIs are defined in Java TV [16].

NOTE: This includes the packages `javax.tv`, `javax.media` and their subpackages.

11.4 Presentation APIs

11.4.1 Graphical User Interface API

11.4.1.1 The Core GUI API

Property names for use with the `getProperty` method on `java.awt.Image` and its sub classes beginning "`dvb.`" are reserved for future use.

Applications shall be able to use Toolkit.beep without any adverse effects to the application. The output is not required to be audible to normal end users and shall not conflict with any other API.

The methods `getScreenResolution` and `getScreenSize` shall be supported with the additional semantics described in HAVi [50].

The encoding of image content types for use by `java.awt.image` are defined in clause 7.1.1, "Bitmap image formats". The set of formats supported is profile dependent.

When using the `java.awt.FontMetrics` class, the width of a set of characters or string returned by the `charsWidth` or `stringWidth` method shall be correct taking into account any kerning and sub-pixel positioning applied by the font renderer. Calculating the same number by adding the widths of the individual characters is not required or expected to return the same number since it will not take into account any kerning or sub-pixel positioning applied by the font renderer.

The downloading of fonts from the network (see clause D.2.2, "Downloaded fonts") shall be supported using the methods concerned on `org.dvb.ui.FontFactory`. Failure to download a font shall be reported by these methods. The constructor for `java.awt.Font` shall only be aware of resident fonts, as described in clause 7.3, "Resident fonts". These fonts are available e.g. to the AWT APIs without using the downloadable font API specified in annex U, "Extended graphics APIs".

For GEM terminals, the time at which an input event occurs as reported by `java.awt.event.InputEvent.getWhen` does not mean the time at which the event is reported to a listener of a GEM application. In particular, if a GEM application has a backlog in processing events and events are being queued, the time at which the event occurred shall be before the event enters the queue and hence also before when it leaves the queue to be reported to the GEM

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 128 of 861

application's listener. The return value of `getWhen()` shall be the difference, measured in milliseconds, between the time when the event occurred and midnight, January 1, 1970 UTC, i.e. like `System.getCurrentTimeMillis()`.

In implementations of GEM, the "single instance of an application-created Frame" that "is permitted per GraphicsDevice" will be created by the GEM implementation for the default `HGraphicsDevice`. Attempts by GEM applications to construct instances of Frame for the default `HGraphicsDevice` will cause the constructor to throw `java.lang.UnsupportedOperationException`.

11.4.1.2 TV user interface

The packages `org.havi.ui` and `org.havi.ui.event` defined in HAVi [50] shall be supported with the following modifications. The following HAVi UI widgets and supporting classes and interfaces are not required by GEM:

- `org.havi.ui.HAnimateEffect`
- `org.havi.ui.HAnimateLook`
- `org.havi.ui.HAnimation`
- `org.havi.ui.HDefaultTextLayoutManager`
- `org.havi.ui.HEventMulticaster`
- `org.havi.ui.HFlatEffectMatte`
- `org.havi.ui.HFlatMatte`
- `org.havi.ui.HGraphicButton`
- `org.havi.ui.HGraphicLook`
- `org.havi.ui.HIcon`
- `org.havi.ui.HImageEffectMatte`
- `org.havi.ui.HImageMatte`
- `org.havi.ui.HListElement`
- `org.havi.ui.HListGroup`
- `org.havi.ui.HListGroupLook`
- `org.havi.ui.HMultilineEntry`
- `org.havi.ui.HMultilineEntryLook`
- `org.havi.ui.HRange`
- `org.havi.ui.HRangeLook`
- `org.havi.ui.HRangeValue`
- `org.havi.ui.HScreenDimension`
- `org.havi.ui.HSinglelineEntry`
- `org.havi.ui.HSinglelineEntryLook`
- `org.havi.ui.HStaticAnimation`
- `org.havi.ui.HStaticIcon`
- `org.havi.ui.HStaticRange`
- `org.havi.ui.HStaticText`

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- org.havi.ui.HSwitchable
- org.havi.ui.HText
- org.havi.ui.HTextButton
- org.havi.ui.HTextLook
- org.havi.ui.HToggleButton
- org.havi.ui.HtoggleGroup

If the optional support for MPEG-2 I-Frames is not included, all attempts to request an instance of org.havi.ui.HStillImageBackgroundConfiguration shall fail. The class itself shall still be present.

The following text shall be considered to be present on the end of the method description of org.havi.ui.HScene.setVisible:

- If this HScene is already visible, then this method brings it to the front. The semantics of show() and setVisible(true) are identical.

  This behaviour is optional in GEM terminals.

Instances of org.havi.ui.event.HRcEvent are reported through the normal java.awt event mechanism due to the inheritance from java.awt.event.KeyEvent.

With the exception of the HSound.load method, no methods specified in HAVi [50] shall throw a security exception in the GEM context. The permissions for HSound.load are those defined for java.io.

The following semantics shall be used for the getVideoController method on HVideoDevice.

- It shall only return JMF players (see javax.media in Java TV [16]) which are in the Prefetched or Started states and which are using that HVideoDevice as one of their scarce resources. Otherwise null will be returned. It shall not return JMF players from other applications if those are using the video device underlying the HVideoDevice.
- Except as specified below, it shall only return JMF players which have been already created in response to the application calling javax.media.Manager.createPlayer or which have been returned by javax.tv.service.selection.ServiceContext.getServiceContentHandlers (see Java TV [16]).
- The only exception to the above is the situation where video is being played in the background as part of the context of an application but where javax.tv.service.selection.ServiceContext.getServiceContentHandlers has not yet been called. In this case, getVideoController shall return the same JMF player as would be returned by getServiceContentHandlers if it was to be called subsequently.

The signatures of the classes HComponent and HContainer shall be extended with "implements org.dvb.ui.TestOpacity".

The methods HGraphicsConfiguration.getPunchThroughToBackgroundColor shall not be used by inter-operable applications.

The methods fontAvailable() and downloadFont() in the class org.havi.ui.HFontCapabilities shall not be used by inter-operable applications.

NOTE 1: The package javax.tv.graphics is required, as per clause 11.3.3, "Java TV".

Applications shall only pass in calls to the method javax.tv.graphics.TVContainer.getRootContainer the exact same XletContext instance as was passed to their initXlet method when it was called by the GEM terminal. The behaviour of getRootContainer if passed any other XletContext is implementation dependent and returning null is one valid option. Except for this case, this method shall never return null in a GEM implementation. The methods javax.microedition.xlet.XletContext.getContainer and javax.tv.graphics.TVContainer.getRootContainer shall return the same as would be returned by a call to org.havi.ui.HSceneFactory.getDefaultHScene() under the same circumstances.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 130 of 861

The method HComponent.isDoubleBuffered shall not be used by inter-operable applications.

Any changes made to the org.havi.ui.HGraphicsConfiguration of an org.havi.ui.HGraphicsDevice shall be reported via the java.awt.GraphicsDevice and java.awt.GraphicsConfiguration APIs, as appropriate. For each HGraphicsConfiguration, there shall be a GraphicsConfiguration where getBounds() returns the same values as HGraphicsConfiguration.getPixelResolution().

NOTE 2: Application developers should not rely on GEM terminals supporting full-screen exclusive mode, thus, as permitted by Personal Basis Profile, the method java.awt.GraphicsDevice.isFullScreenSupported may always return false.

The system property dvb.display.aspect_ratio shall indicate the shape of the physical display that the GEM terminal believes is used by its default HScreen.This may not be the same as the aspect ratio of the receiver's current output signal. This shall be encoded as the ratio of width to height, both expressed as positive decimal integers and separated by a ":", e.g. "4:3" or "16:9". This shall be the same ratio as would be returned by VideoFormatControl.getDisplayAspectRatio for a JMF player decoding on that HScreen.

11.4.1.3 Extended graphics

See annex U, "Extended graphics APIs".

The class org.dvb.ui.DVBTextLayoutManager is not required by the present document.

11.4.1.4 Television viewing mode

GEM includes two ways for applications to receive input events: the normal AWT method in java.awt.Component and the org.dvb.event package (see clause 11.3.2.2, "org.dvb.event"). Via the normal AWT method, the application normally receives all input events when the component has the focus. The minimum set of input events that are supported is defined in clause G.5, "Input events".

Often the resident navigator software of the GEM terminal uses many of the keys on the remote control for its own navigation purposes. The navigator shall not act on input events which are delivered to GEM applications. In particular, the navigator shall not respond to number key input if that input is delivered to one or more GEM applications. Conversely, if the GEM application with input focus has indicated disinterest in number key input (using HScene.setKeyEvents with an HEventGroup not including the number keys) then navigators which respond to number key input are free to do so. Rules on when GEM applications should request interest in particular input events and when they should not are found below.

To ensure consistent user experience, the following rules are defined:

- An application creating an HScene and placing components into it shall not by default get the input focus for these components.
- The application may request to get the input focus by calling Component.requestFocus(). If this is granted and the focus moved to the requested component, this component shall receive input events as defined in clause J.1.
- The application may request to receive a subset of input events via the org.dvb.event API even when not having the AWT focus.

For applications delivered within normal television services, it is recommended that the following items are taken into account:

- When the display is primarily showing the video of the television service and the user perception is that he is not actively interacting with an application but is just watching the television service (called 'television viewing mode'), the applications should not request the AWT focus but let most of the input events go to the resident navigator (e.g. number keys, directional arrow keys and Enter may cause the normal actions the user expects from the navigator).
- In the 'television viewing mode' the applications should request only the minimum required input events, e.g. only input events that cause some further action to happen and that may transition the user from the 'television viewing mode' into a state where the user more actively interacts with the applications.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- If the applications require some input events for the user to be able to activate the application when in the 'television viewing mode', it is recommended that the VK_COLORED_KEY_(0 to 3) and VK_TELETEXT (see note below) input events are used for this purpose and the applications request them via the org.dvb.event API. In practice this means that the navigator can not map any essential function directly to the VK_COLORED_KEY(0 to 3) events in television viewing mode.
- Subtitles should be available when end-users perceive themselves as watching television. In order to achieve this, GEM applications shall not have a visible full screen HScene where more than 60% of the pixels are wholly transparent to video when in "television viewing mode".
- GEM applications wishing to leave "television viewing mode" and receive events other than the VK_COLORED_KEY_(0 to 3) and VK_TELETEXT are obliged to display a non-transparent graphics object on the screen that covers at least 3 % of the visible area on the screen (equivalent to 9 852 pixels on a 634 x 518 visible screen in a 720 x 576 or 720 x 480 graphics mode) which clearly indicates to the user the "non-navigator" mode of the receiver with respect to input event handling. (Obviously, the pixel values given only apply to systems with the indicated resolution; on other systems, the 3 % requirement applies, but results in different pixel values.)

NOTE 1: In television services where DVB Teletext is transmitted within the service, a typical navigator action in the 'television viewing mode' can be to activate a teletext decoder, if supported by the terminal. The application requesting this input event may cause the DVB Teletext decoder not to be conveniently accessible to the end user. Therefore, within services that carry DVB Teletext it is recommended that the VK_TELETEXT input event is requested in the 'television viewing mode' only by such applications that provide a simulcast GEM application version of the teletext service, thus providing a replacement for the DVB Teletext information.

In environments where WST Teletext services are prevalent when there is no DVB Teletext service associated with the video service, the GEM application should display an information service of a similar nature to teletext. When receiving this event for the second time the information service should terminate and the user will return to normal viewing mode.

The VK_TELETEXT event should have no other purposes in GEM applications.

NOTE 2: The items above apply only to the 'television viewing mode' as explained above. In states where the user experience is that the user is clearly primarily interacting actively with an application the applications can naturally use the normal AWT input event mechanism and request the focus as well as use any of the supported input events via the org.dvb.event API.

NOTE 3: For data services, the first interaction state of an application that is automatically started after service selection should behave as television viewing mode. This is in order to allow users who are navigating with the navigator to seamlessly navigate through these services without getting confused.

NOTE 4: As a consequence of clause 4.1.4, "Addition of non-GEM interfaces" the requirements of this clause are required of all GEM-based terminal specifications; as a further consequence, GEM terminal specifications are not permitted to define extensions that have to be invoked by applications in order to obtain the behaviour mandated by the present document.

NOTE 5: The method java.awt.event.KeyEvent.getKeyChar() may return java.awt.event.KeyEvent.CHAR_UNDEFINED only for KEY_PRESSED and KEY_RELEASED events, never for KEY_TYPED events. In no case should an interoperable application rely on the return value of getKeyChar() for a KEY_PRESSED or KEY_RELEASED event; further, the value returned for an undefined key is not defined interoperably and may vary across platform implementations.

11.4.1.5 Font bindings

11.4.1.5.1 PFR0

For fonts in the PFR0 format clause 7.4.1, "PFR", the bindings between the Java APIs relating to font metrics and the font format itself shall be as follows;

The return values of the methods FontMetrics.getMaxAscent and FontMetrics.getMaxDescent shall be yOffsetTop and yOffsetBottom respectively as defined in clause D.3.5.3, "Conversion of units".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 132 of 861

For PFR fonts with the auxiliary data record:

- `FontMetrics.getAscent` shall be obtained from the `ascent` field.
- `FontMetrics.getDescent` shall be obtained from the `descent` field.
- `FontMetrics.getLeading` shall be obtained from `externalLeading`.

  All values in the auxiliary data record shall be converted from font metrics units to pixels as described in clause D.3.4, "Converting font metrics to display pixels" before being returned. This conversion shall round up.

For PFR fonts without the auxiliary data record:

- `FontMetrics.getAscent` shall be the same as `getMaxAscent`.
- `FontMetrics.getDescent` shall be the same as `getMaxDescent`.
- `FontMetrics.getLeading` shall be zero.

The "advance width" of a character, in metrics units, is defined as follows:

- For PFR fonts with the "`proportionalEscapement`" flag set, the advance width of a character is given by the "`charSetWidth`" field in the character record.
- For PFR fonts without the "`proportionalEscapement`" flag set, the advance width of a character is given by the "`standardSetWidth`" field in the physical font record.

`java.awt.FontMetrics.charWidth()` shall return the advance width of the character converted to pixels. This conversion shall round up.

`java.awt.FontMetrics.stringWidth()`, `java.awt.FontMetrics.bytesWidth()` and `java.awt.FontMetrics.charsWidth()` are calculated by summing the advance widths of all the characters in the string, and adjusting for any kerning in the font. Adjustments for kerning are performed in metrics units, then the result is converted to pixel units. This conversion shall round up.

`java.awt.FontMetrics.getMaxAdvance()` returns the maximum value that `java.awt.FontMetrics.charWidth()` can return.

NOTE: For proportional fonts the implementation calculates this value from the advance width of every character in the font; there is no field in the PFR file which contains it.

11.4.1.5.2 OpenType

For fonts in the OpenType format as described in clause 7.4.2, "OpenType", the binding between the Java API relating to font metrics and the font format itself shall be as follows.

The return values of the methods `FontMetrics.getMaxAscent`, `FontMetrics.getMaxDescent`, `FontMetrics.getAscent`, `FontMetrics.getDescent and FontMetrics.getLeading` shall be as follows:

- `FontMetrics.getMaxAscent` shall be obtained from the `yMax` field of the `Font Header ('head')` table;
- `FontMetrics.getMaxDescent` shall be obtained from the `yMin` field of the `Font Header ('head')` table, and shall return `-yMin`;
- `FontMetrics.getAscent` shall be obtained from the `sTypoAscender` field in the platform-independent OS/2 tables.
- `FontMetrics.getDescent` shall be obtained from the `sTypoLineGap` field in the platform-independent OS/2 tables.
- `FontMetrics.getLeading` shall be obtained from the `LineGap` field in the `Horizontal Header ('hhea')` table.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 133 of 861

The results of `java.awt.FontMetrics.stringWidth()`, `java.awt.FontMetrics.bytesWidth()` and `java.awt.FontMetrics.charsWidth()` return the total advance width for rendering the specified object and may account for advanced typesetting features.

NOTE: The total advance width returned from these methods is affected by the layout engine, which might use advanced typesetting features like kerning or ligatures. The advance of a sequence of characters is not necessarily the sum of the advances of its characters.

11.4.2 Streamed Media API

11.4.2.1 Framework of solution

The `javax.media` and `javax.media.protocol` packages from Java TV [16] shall be implemented with the clarifications, extensions and restrictions as defined in the corresponding clauses below.

11.4.2.2 Clarifications

The JMF "time base time" when playing MPEG content delivered in MPEG transport streams is used as a constantly progressing time whose rate may be synthesized from the Program Clock References / the System Time Clock in MPEG or by some other appropriate method. The value does not have any direct relation to the value of the MPEG System Time Clock in the receiver.

The media time of JMF when playing MPEG content delivered in MPEG transport streams is just a time value that progresses as the media stream is played, but whose actual value is implementation dependent.

NOTE: There is no implied or expected connection between the JMF media time and any time base(s) provided by the MPEG-2 / DSM-CC Normal Play Time.

When creating a JMF player, Media locators and URLs which reference a service, event or elementary stream will create a player which plays the content concerned direct from the network. Media locators and URLs which reference files will create a player which will download the content concerned from the network during the Prefetching state of the player concerned. If the content cannot be downloaded then such players will never enter the Prefetched state.

Interoperable applications shall not call `javax.media.MediaHandler.setSource()`.

In the `javax.media.PackageManager`:

a) The effect of the `set<xxx>PrefixList` methods is limited to the application calling the method.

b) The `SecurityException` of the `commit<xxx>` methods shall always be thrown in GEM.

The events `javax.media.ControllerEvent` and `javax.media.GainChangeEvent` shall inherit from `java.util.EventObject`.

The `interfaces javax.media.ControllerListener` and `javax.media.GainChangeListener` shall inherit from `java.util.EventListener`.

`SubtitlingLanguageControl.isSubtitlingOn()` returns the current state of subtitle presentation and NOT the last value set with `setSubtitling`. In particular `setSubtitling` may be set to true but if there are no subtitles in the network or no subtitle service component selected then `isSubtitlingOn` shall return false.

Any elements in `DVBLocators` after and including the `event_id` element are ignored by JMF players.

NOTE: For GEM terminal specifications that do not include the functional equivalent named "Content Referencing" as defined in clause 15.6, "Functional equivalents", this paragraph does not apply. Instead, the present document requires that any information in a locator beyond that identifying a service (e.g. the time of a specific program event) is to be ignored by JMF players. See also clause 14.8, "Locators and content referencing".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.4.2.3 Default media player behaviour

For a JMF player which is presenting a service, the following rules will be followed in the order given to decide which service components will be presented when multiple audio, video or subtitle components are present in a service:

a) For audio and subtitle streams in different languages, user preferences will be used to determine which streams are selected.

b) The streams which are first in the network signalling information (i.e. in the PMT) will be presented. GEM terminal specifications may define additional mechanisms that override this default.

If any of the media components comprising the service change then the implementation will as far as possible replace the changed components with suitable replacements in a user preference and implementation dependent manner.

11.4.2.4 Required controls for video drips

The following controls are supported for GEM terminal specifications that support video drips (see clause 7.1.3, "MPEG-2 Video "drips""):

- `javax.tv.media.AWTVideoSizeControl`
- `org.dvb.media.BackgroundVideoPresentationControl`

11.4.2.5 Extensions to the Framework

11.4.2.5.1 DVB specified extensions

The classes and interfaces defined in annex N, "Streamed media API extensions" of the present document are included, with the exception of the following:

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

If a `Player` bound to a `DripFeedDataSource` receives a `ResourceWithdrawnEvent` and later a `ResourceReturnedEvent`, the video output from the video decoder to which the player is bound is implementation dependent. The GEM application using the `Player` shall call the `DripFeedDataSource.feed` method with an I-frame as soon as possible after `ResourceReturnedEvent` is received.

NOTE: Application developers should take this into account when constructing video drip sequences. A large number of P-frames between I-frames could lead to a significant delay before the visible display can be refreshed due to the limitations on the frequency with which new data can be fed to the decoder.

11.4.2.5.2 Extensions in org.davic

The following classes and interfaces will be included from the `org.davic.media` package, as defined in annex L of DAVIC 1.4.1p9 [17].

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- MediaPresentedEvent
- MediaLocator
- MediaTimePositionControl
- ResourceWithdrawnEvent
- FreezeControl
- ResourceReturnedEvent
- MediaTimePositionChangedEvent
- LanguageNotAvailableException
- NotAuthorizedException
- MediaFreezeException
- LanguageControl
- AudioLanguageControl

For the packaged media target, the following classes are not required by the present document:

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

The following class is required for terminal specifications that include the conditional access API:

- org.davic.media.NotAuthorizedMediaException

The following classes will be included from the org.davic.media package as defined in annex L of DAVIC 1.4.1p9 [17] with the following semantic modification:

- The completion of the action started by calling setMediaTimePosition() on the MediaTimePositionControl will be signalled by a org.davic.media.MediaTimePositionChangedEvent.
- The org.davic.media.MediaPresentedEvent shall only be thrown following changes caused using controls in the org.davic package and as a consequence of the transition of a JMF player into the Started state. Instances of this event shall be generated as close to the presentation of the content as the device can detect, e.g. when the new content leaves the output of a hardware decoder. In this second case, it augments the JMF state transition events and does not replace them.

NOTE 1: Changes effected using javax.tv.media.MediaSelectControl do not lead to the org.davic.media.MediaPresentedEvent being thrown.

NOTE 2: The present document defines more specific conditions under which ResourceWithdrawnEvent and ResourceReturnedEvent than the original DAVIC specification. This can be found in clause 11.6.2, "Service selection API".

11.4.2.5.3 Extensions in javax.tv

In javax.tv.media.MediaSelectControl the InsufficientResourcesException is thrown by the select() method if selecting any service component fails due to a lack of system resources.

In javax.tv.media.AWTVideoSize.equals(), the "data members" shall be considered to be the "source" and "dest" parameters used to create the object.

The components returned by javax.tv.media.MediaSelectControl.getCurrentSelection are the currently selected components which may be different from those previously selected by this control. Mechanisms which may modify the current selection include:

- Control of subtitles and audio language.
- The CA system including the common interface e.g Host Control replace / clear_replace.
- User intervention via the navigator.

The following clarifications shall apply to the method getDefaultSize in javax.tv.media.AWTVideoSizeControl:

a) This method shall return the current default AWTVideoSize. The default shall be that which would be implemented by the GEM terminal when the video scaling and positioning of a JMF player is under control of the GEM platform (see org.dvb.media.VideoFormatControl.DFC_PLATFORM). When the video scaling

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

and positioning is in the DFC_PLATFORM mode, the return value of this method shall change to track changes made by the policy underlying DFC_PLATFORM.

b) Calling AWTVideoSizeControl.setSize() with the AWTVideoSize object returned by this method as the parameter shall set the video scaling to the current default at the time getDefaultSize() was called. Hence it shall be equivalent to calling BackgroundVideoPresentationControl.setVideoTransformation with the video transformation returned by the method VideoFormatControl.getVideoTransformation(getDecoderFormatConversion()) and not with the video transformation returned by getVideoTransformation(DFC_PLAFORM).

NOTE: When the GEM terminal is in pan and scan mode, (see org.dvb.media.VideoFormatControl.DFC_PROCESSING_PAN_SCAN), the return value of AWTVideoSizeControl.getSize() will be out of date almost as soon as the method has returned.

11.4.2.5.3.1 Enumerated Types

Java TV defines the following enumerated types:

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

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

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

11.4.2.5.4 Required controls for broadcast profiles and packaged media profiles

The following controls are supported for the broadcast streaming formats specified in clause 7.2, "Media streaming formats":

- org.davic.media.LanguageControl
- org.davic.media.AudioLanguageControl
- org.davic.media.SubtitlingLanguageControl
- org.davic.media.FreezeControl
- javax.tv.media.MediaSelectControl
- javax.tv.media.AWTVideoSizeControl
- org.dvb.media.VideoPresentationControl
- org.dvb.media.BackgroundVideoPresentationControl
- org.dvb.media.SubtitlingEventControl
- org.dvb.media.VideoFormatControl

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- `org.dvb.media.DVBMediaSelectControl`

The following controls are supported for media decoded from clause 7.1.4, "Monomedia format for audio clips":

- `org.davic.media.MediaTimePositionControl`

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

11.4.2.5.5 Clarifications

In `SubtitlingLanguageControl`, subtitling being presented shall be defined as the subtitle decoder running. It does not require subtitles to be visible on screen at that time. See clause 13.5, "Subtitles".

When a `PresentationChangedEvent` is generated, the Player in question should re-evaluate the list of controls returned by the `getControls()` method for that player. Similarly, Players supporting the `MediaSelectControl` should re-evaluate the list of Controls returned by `getControls()` after calls to the `MediaSelectControl.select()` method is called.

Any controls referenced by an application after a `PresentationChangedEvent` or a `MediaSelectSucceededEvent` is generated may fail silently if they are no longer valid. Controls which are valid for the new content remain unaffected. Applications should re-acquire any controls they require after either of these events has been generated.

If the content being presented by a JMF player becomes unavailable due to tuning, `org.davic.media.ResourceWithdrawnEvent` shall be sent to listeners registered to receive that events. A `ResourceReturnedEvent` shall be sent to all listeners registered at that time if the content becomes available again.

The following controls shall work on a JMF Player in any state (including `Unrealized`, `Realizing`, `Realized`, `Prefetching`, `Prefetched` and `Started` Players) and changes made via these controls shall be preserved across JMF state transitions:

- `javax.tv.media.AWTVideoSizeControl`
- `org.dvb.media.VideoPresentationControl`
- `org.dvb.media.BackgroundVideoPresentationControl`

It is implementation-specific whether or not the effect of other JMF controls included in the present document survives JMF state transitions.

When decoding of the media stream is frozen by `org.davic.media.FreezeControl`, scaling the video (e.g. by `BackgroundVideoPresentationControl`) it is recommended that this have immediate effect on the existing frozen video still. GEM terminals that cannot do this shall temporarily resume decoding of the media stream to acquire a new video still that shall be scaled as specified, and shall automatically re-freeze the media after this video still has been acquired.

Where streaming service components of the same service (e.g. MPEG-2 elementary streams) are being decoded by different JMF players, there shall be no synchronisation between those components. Specifically, if the video and audio from a service are being presented by separate JMF players then they shall be presented in "free-running" mode ignoring the PCR.

11.4.2.5.6 Component-based JMF players

JMF Players start as background players. Players which do not provide a `org.dvb.media.ComponentBasedPlayerControl` can only be used to present video in the background and their `getVisualComponent` method shall always return null. Players that provide an implementation of the `org.dvb.media.ComponentBasedPlayerControl` are capable of acting both as background and component-based players, and their `getVisualComponent` method must follow the semantics defined by that control and this clause.

If an application calls `Player.getVisualComponent()` and receives a non-null Component, that application then owns the visual component for that Player. It continues to own the visual component for that Player until the application is destroyed or calls `ComponentBasedPlayerControl.releaseVisualComponent()`. An application that owns

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

the visual component of a Player will get the same component returned by calls to `Player.getVisualComponent()` on that Player until it releases the component. Calls to `Player.getVisualComponent()` will return null if the visual component for that Player is owned by another application.

If `getVisualComponent` returns non-null, video shall continue running in the background until the first call to the paint method of the component returned by `getVisualComponent`. When this transition happens the video is resized and repositioned as required by the visual component. Possible areas outside of the component are updated to follow the rules in clause 13.3, "Graphics".

After this transition, applications shall re-acquire the list of JMF controls for the player. It is implementation dependent whether any previous JMF controls are re-used. Applications shall not use any previous JMF controls which are not in the new list of JMF controls. When a JMF player is performing as a component based player, the following JMF controls shall not be supported and video scaling and positioning shall be done by controlling the size and position of the component returned by the `getVisualComponent` method.

- `org.dvb.media.BackgroundVideoPresentationControl.`
- `javax.tv.media.AWTVideoSizeControl.`

Players which return a non-null `java.awt.Component` from the `getVisualComponent` method must fully support the semantics for `java.awt.Component` concerning positioning and scaling.

11.4.2.5.7 Streaming Monitoring API

The Streaming Monitoring API is required in the OTT target. The classes and interfaces are defined in annex "N.3 Streaming Monitoring API" of the present document.

The Streaming monitoring API is used to expose to applications information about streamed CoD played by JMF Player. This API is applicable with Adaptive Streaming as well as to non-adaptive streams and allows an application to be notified about changes in buffer status and quality of played stream.

The API is composed from the following classes and interfaces:

- `MonitoringControl`
- `MonitoringEvent`
- `StreamQualityChangeEvent`
- `StreamQualityInfo`
- `VideoStreamQualityInfo`
- `BufferingStatusEvent`
- `MonitoringListener`

The `MonitoringControl` is a new `Control` and its implementation shall be returned by JMF `Player` (`javax.media.Controller.getControl()`, `javax.media.Controller.getControls()`) used to play OTT media streams independently from whether the system supports adaptive streaming. By using this `Control` an application has access to information about played stream. Application is able to check audio and video format, current quality of a stream being presented to the user as well as all available qualities describes in a manifest file passed as a Locator to the Player, current download bitrate of a stream and line rate. It is also able to set a quality of the associated CoD.

Manifest file tells the receiver that several alternative representations are available on the server for the same content, e.g. more files corresponding to encodings at different bit rates of the same content which the receiver can seamlessly switch to based on varying network conditions.

`MonitoringEvent` is a base class for any event related to changes in the properties of playing media. At the current state `StreamQualityChangeEvent and BufferingStatusEvent` can occur.

BufferingStatusEvent shall be used when some changes in buffering status appear, e.g. buffer is underrun or network connection has been lost and buffer is not filling anymore. To prevent implementations from being overloaded buffering status events should not be sent faster than one per seconds.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 139 of 861

`StreamQualityChangeEvent` shall be used when JMF Player is switching to another stream defined in media's manifest file due to changes in network conditions (Adaptive Streaming).

`StreamQualityInfo` is a class used to encapsulate necessary information about each stream described in media's manifest file.

`VideoStreamQualityInfo` is a subclass of `StreamQualityInfo` class. It is dedicated to encapsulate information specific for video streams.

To receive `MonitoringEvent` events, an object has to implement the `MonitoringListener` interface and use `MonitoringControl.addMonitoringListener()` method to register its interest in media property's events. All events are posted to each registered listener.

11.4.2.5.8 Media Stream Synchronization API

The media stream synchronization API is an optional extension of the GEM core APIs, which enables to establish a synchronization link between multiple JMF players. It enhances the JMF player model as defined in JavaTV [16], which already defines a limited master/slave concept (See the section "managing controllers" in class javax.media.Player in JavaTV for more details). For this purpose a set of new controls and events are defined, which provides a flexible master/slave extension which is in-line with the JavaTV architecture.

The media stream synchronization API is defined in "N.3 Media Stream Synchronization API", an informative usage example is in Annex W.5.

The media stream synchronization API can be deployed on any existing and future GEM targets. It may be used to synchronize streams on terminals deployed only on a single network (e.g. dual tuner STBs) as well as on hybrid terminals, on which the GEM targets are combined into a hybrid profile.

The media stream synchronization API can be used to couple JMF players into a master and slave relationship and enables synchronization of the media presentation by synchronizing the clocks of these players. The media presentation can be controlled by controlling the master player, the slave player is synchronized accordingly. A master/slave relationship between JMF players lasts until the slave is explicitly removed from the master. Any JMF player becomes a master by adding a slave to it. Any JMF player becomes a slave, when it is added as a slave to another player.

Changes to the presentation state of the master player change the state of the slave player accordingly, i.e. starting a master player automatically starts the slave. Additionally all methods and controls, which change the media time or rate of the master player also have the same effect on the slave. Note that the slave can still be controlled directly, i.e. it can be moved to a different state (e.g. stopped) or set to a different media time. Any state change on a slave player doesn't affect the state of the master player. It is not permitted to set the media time and media rate of a slave player to a value different from the master player.

11.4.2.5.8.1 API behavior in border cases

If an application calls a method with a null parameter instead of an object instance, the method throws a NullPointerException. Methods which return an array of elements return an empty array if no element is available.

11.4.2.5.8.2 Establishing a Master/Slave Relationship

A master/slave relationship is a temporary synchronization relationship between JMF players. A player obtains the master role, i.e. it contains the reference clock to which other slave players can be synchronized. The master/slave relationship is established dynamically by adding a player as a slave to another player. It lasts until the slave is removed. The API permits to add multiple slaves to a master player. When a player has a slave role, it cannot act as a master for other players, i.e. there's only a one-level master/slave hierarchy.

An application, which wants to establish a master/slave relationship among players, can obtain an instance of `MasterSlaveSyncLinkageControl` of the master player. On platforms which don't allow synchronization, a call to `getControl("org.dvb.media.MasterSlaveSyncLinkageControl")` on any player will return null.

A master/slave relationship is established when another player is added to the master via the `addSlave` method of the `MasterSlaveSyncLinkageControl`, which becomes a slave player when the `addSlave` method was

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 140 of 861

successfully executed. A SyncControl can be obtained for any player, which can be used as a slave player but the methods of `SyncControl` are only effective, while a Master/Slave relationship is established.

To remove a slave from a master/slave relationship, the method `removeSlave` is used. If a slave had been removed, it can be controlled independently from the master. After all slaves are removed from a master, the master is again a normal player and could serve as a slave to another master.

Adding or removing a slave player can happen in any state of the master and slave player. These operations don't impact the player states of the master and the slave, i.e. the return value of `Player.getState()` does not change for the master and slaves.

Not all media streams and their corresponding JMF players can be successfully synchronized - If synchronization can never be established during the `addSlave` method, an `IncompatibleSynchronizableMediaTypeException` is thrown.

11.4.2.5.8.3 Adding a Slave

When a slave has been added, the master `Player`:

- Invokes `setTimeBase` on the `slave` with the master `Player's TimeBase`. If this fails, `addSlave` throws an `IncompatibleTimeBaseException`.
- Synchronizes the slave with the `Player` using `setMediaTime`, `setStopTime`, and `setRate`.
- Takes the added slaves latency into account when computing the master `Player's` start latency. When `getStartLatency` is called, the master `Player` returns the greater of: its latency before the `Controller` was added and the latency of the added `Controller`.
- Takes the added slave's duration into account when computing the master `Player's` duration. When `getDuration` is called, the `Player` returns the greater of: its duration before the `slave` was added and the duration of the added `slave`. If either of these values is DURATION_UNKNOWN, `getDuration` returns DURATION_UNKNOWN. If either of these values is DURATION_UNBOUNDED `getDuration` returns DURATION_UNBOUNDED.
- Adds itself as a `ControllerListener` for the added slave so that it can manage the events that the slave generates. (See the Events section below for more information.)
- Invokes control methods on the added slave in response to methods invoked on the master `Player`. The methods that affect slave players are discussed below.

11.4.2.6.8.4 Removing a Slave

When a `slave` is removed from a master `Player's` control, the master `Player`:

- Resets the slave's `TimeBase` to its default.
- Recalculates its duration and posts a `DurationUpdateEvent` if the master `Player's` duration is changed.
- Recalculates its start latency.

11.4.2.5.8.5 Starting a Master Player

In accordance with the behavior for managed players as described in JavaTV, the start behavior is as follows:

When you call `start` on a master `Player`, all of the `slaves` managed by the `Player` are transitioned to the *Prefetched* state. When the `slaves` are *Prefetched*, the master `Player` calls `syncStart` with a time consistent with the latencies of each of the slaves.

Calling realize, prefetch, stop, or deallocate on a Master Player
When you call `realize`, `prefetch`, `stop`, or `deallocate` on a master `Player`, the `Player` calls that method on all of the `slaves` that it is managing. The `Player` moves from one state to the next when all of its `slaves` have reached that state. For example, a `Player` in the *Prefetching* state does not transition into the *Prefetched* state until all of its `slaves` are *Prefetched*. The `Player` posts `TransitionEvents` normally as it changes state.

Calling syncStart or setStopTime on a Master Player

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

When you call `syncStart` or `setStopTime` on a master `Player`, the `Player` calls that method on all of the `slaves` that it is managing. (The `Player` must be in the correct state or an error is thrown. For example, the `Player` must be *Prefetched* before you can call `syncStart`.)

Setting the Time Base of a Master Player
When `setTimeBase` is called on a master `Player`, the `Player` calls `setTimeBase` on all of the slaves it's managing. If `setTimeBase` fails on any of the `Controllers`, an `IncompatibleTimeBaseException` is thrown and the `TimeBase` last used is restored for all of the `Controllers`.

Getting the Duration of a Master Player
Calling `getDuration` on a master `Player` returns the maximum duration of all of the added `Controllers` and the master `Player`. If the `Player` or any slave has not resolved its duration, `getDuration` returns `Duration.DURATION_UNKNOWN`.

Closing a Master Player
When `close` is called on a master `Player` all slaves are closed as well.

11.4.2.5.8.6 Setting the Media Time and Rate of a Master Player

In accordance with the behavior for managed players as described in JavaTV, the event handling is as follows:

When you call `setMediaTime` on a master `Player`, its actions differ depending on whether or not the `Player` is *Started*. If the master `Player` is not *Started*, it simply invokes `setMediaTime` on all of the slaves it's managing. If the `Player` is *Started*, it posts a `RestartingEvent` and performs the following tasks for each slave:

- Invokes `stop` on the slave.
- Invokes `setMediaTime` on the slave.
- Invokes `prefetch` on the slave.
- Waits for a `PrefetchCompleteEvent` from the slave.
- Invokes `syncStart` on the slave

The same is true when `setRate` is called on a master `Player`. The `Player` attempts to set the specified rate on all slaves, stopping and restarting the `Controllers` if necessary. If some of the `Controllers` do not support the requested rate, the `Player` returns the rate that was actually set. All `Controllers` are guaranteed to have been successfully set to the rate returned.

11.4.2.5.8.7 Loss of Synchronization

The API permits to set a synchronization tolerance time value, within which the different Media streams may deviate from the master. If the master/slave player cannot keep the players synchronized within the tolerance value, this condition is reported with a `SyncStateChangedEvent`. After the synchronization is re-established between the master and the slave, this is also reported with a `SyncStateChangedEvent`.

A slave player always attempts to resynchronize to the master, however when it determines that synchronization will never be established, it set its status to unsynchronizable and reports this with an SyncStateChangedEvent with unsynchronzable.

11.4.2.5.8.8 Event Handling

In accordance with the event behavior for managed players as described in JavaTV, the event handling is as follows:

Most events posted by a slave are filtered by the master `Player`. Certain events are sent directly from the slave through the master and to the listeners registered with the master.
To handle the events that a slave can generate, the master registers a listener with the slave when the master/slave relationship is established. Other listeners that are registered with the slave are still notified, while it is being managed by the master. When a slave is removed from the master `Player`'s list of slaves, the slave removes itself from the masters listener list.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Transition Events

A master `Player` posts `TransitionEvents` normally as it moves between states, but the slaves affect when the master changes state. In general, a master `Player` does not post a transition event until all of its slaves have posted the event.

Status Change Events

The master `Player` collects the `RateChangeEvents`, `StopTimeChangeEvents`, and `MediaTimeSetEvents` posted by its slaves and posts a single event for the group.

DurationUpdateEvent
A JMF `Player` posts a `DurationUpdateEvent` when it determines its duration or its duration changes. A master `Player's` duration might change if a slave updates or discovers its duration. In general, if a slave posts a `DurationUpdateEvent` and the new duration changes the master's duration, the master Player posts a `DurationUpdateEvent`

CachingControlEvent
A master `Player` reposts `CachingControlEvents` received from a `Players` that it manages, but otherwise ignores the events.

ControllerErrorEvents
A master `Player` immediately reposts any `ControllerErrorEvent` received from a `slave` that it is managing to its `ControllerListener`. After a `ControllerErrorEvent` has been received from a slave, a master `Player` no longer invokes any methods on the slave; the slave is ignored from that point on and the slave is removed from the master.

11.4.2.6 Restrictions on the Framework for Broadcast

Controls that are supported by GEM need not have an associated GUI component. Any calls to `Control.getControlComponent()` may return null.

Developers of GEM applications should not rely on the presence of the following classes or interfaces:

- `javax.media.CachingControl` (unless returned by a call to a JMF method).
- `javax.media.CachingControlEvent`.
- `javax.media.GainControl` (unless returned by a call to a JMF method).

A GEM implementation need not return a `CachingControl` or `GainControl` from a call to `Player.getControls()` however this is not prohibited. If a GEM implementation does return a `GainControl`, then the volume that is set using this control may not be greater than the system volume level. I.e. the gain control may only change the volume between mute and the volume level at the current system volume level. This system volume level shall be represented as 1.0.

If the visual component returned by `Player.getVisualComponent()` is hidden (as returned by `Component.isHidden()`), the video is hidden. It is legal to remove a visual component from an AWT hierarchy and add it back to an AWT hierarchy (perhaps in a different place). The video will be hidden when the component is removed and shown again when the component is painted again.

Applications should not expect `PushDataSource` and `PullDataSource` to exist for broadcast MPEG content, and it is not required that functional implementations be provided for implementations of JMF in DVB. The `DataSource` used may be implementation-specific and non-specified apart from its inheritance from `javax.media.protocol.DataSource`.

The constructor for `URLDataSource` may always throw `IOException` for broadcast MPEG content.

For the purposes of integration with `javax.tv.media.MediaSelectControl` "resynchronization" is not considered to occur provided that the same PCR_PID is referenced between the new and existing service components.

Implementations of JMF shall not tune but shall allow selection of media components from transport streams which can be reached without tuning even where they are not part of a currently selected service.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 143 of 861

The failure modes if a locator is used which cannot be reached without tuning are defined as follows:

- For a JMF player which is a `ServiceMediaHandler`, the failure modes are fully specified in the description of `MediaSelectControl`.
- If a JMF player which is not a `ServiceMediaHandler` is created with such a locator then it shall not be able to enter the `Realized` state.

  The `Realizing` state shall be exited with the posting of a `ResourceUnavailableEvent`.
- For JMF players which are not `ServiceMediaHandlers`, `MediaSelectControl` shall be restricted to operating only on service components belonging to transport streams currently available without tuning.

NOTE: In this version of the present document, no media types are defined for which `javax.media.Player.addController` can have any other behaviour apart from throwing a `IncompatibleTimeBaseException`.

11.4.2.7 Intersection Between MediaSelectControl and SubtitlingLanguageControl/ AudioLanguageControl

The method `org.davic.media.LanguageControl.listAvailableLanguages()` shall list all available languages in the service concerned. This includes languages carried in service components which are not currently selected, either because the service component concerned has been de-selected using `MediaSelectControl` or because it was never selected initially because it was not carried in one of the components listed in the component tags of a locator which included component tags.

For the purposes of the present document, "component tags of a locator" is to be interpreted as meaning the description of the required components in a locator. This clause also places semantic requirements on subtitling-related APIs. As these APIs are optional in the present document, these requirements apply only if these APIs are required by the GEM terminal specification.

The method `org.davic.media.LanguageControl.selectLanguage()` shall over-ride any selection of service components made either by `javax.tv.media.MediaSelectControl` or by using a locator for a service which included component tags. If `selectLanguage()` has changed the set of service components selected, the methods on `javax.tv.media.MediaSelectControl` shall behave as if that change had been made through `MediaSelectControl` and return the correct set of service components taking into account the change made by `selectLanguage()`.

If the methods on `javax.tv.media.MediaSelectControl` are used to replace or remove the service component carrying the currently active audio or subtitles then the presentation of this service component shall stop. If the methods on `javax.media.MediaSelectControl` are used to add a service component carrying audio or subtitles when a service component of this media type is not already being presented then presentation of this service component shall be started. The language of the newly selected service component shall be returned by calls from the application to the method `getCurrentLanguage` on `AudioLanguageControl` or `SubtitlingLanguageControl` respectively. Attempting to select multiple service components of the same media type to be presented at the same time using methods on `MediaSelectControl` shall fail unless the GEM terminal concerned has sufficient resources to present them simultaneously. The failure mode shall be that as specified in `MediaSelectControl` for when insufficient resources are available.

Control of the availability of subtitles using `MediaSelectControl` or the methods `selectLanguage` and `selectDefaultLanguage` (which `SubtitlingLanguageControl` inherits from `LanguageControl`) shall not impact the value of "Application Control" as mentioned in figure 34, "Determining subtitling language and presentation setting" and vice versa.

11.4.2.8 Intersection between Streamed Media API and TV User Interface API

11.4.2.8.1 Basic Principles

The receiver's output picture shape and/or WSS signalling is normally derived from the configuration of the `HVideoDevice` and the decoder format conversion being applied, as in "TV behaviour control". When an application

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

has the HGraphicsDevice reserved, the picture shape and/or WSS signalling shall be derived solely from the output aspect ratio of the HGraphicsDevice.

Some of the decoder format controls pre-defined as part of org.dvb.media.VideoFormatControl may require control over the pixel aspect ratio of the final video output signal after video, graphics and backgrounds have been combined as shown in figure 17. In order to enable this, a JMF player shall attempt to reserve the HVideoDevice instance on which its output is being displayed. Failure to reserve or appropriately configure the HVideoDevice shall not be considered fatal to the JMF player but may result in an inferior TV video presentation.

NOTE: The actual configurations of the HVideoDevice used and available are dependent on the precise configuration of the GEM terminal. Differences will exist between set-top boxes, integrated digital TVs and PCs. The precise nature of any "inferior TV video presentation" is dependent on the configuration of the GEM terminal, on the input video and the characteristics of the final output device. Typically, the result will be distorted, cropped or scaled video.

Component-based Players shall not have an associated HVideoDevice. Therefore the HVideoComponent.getVideoDevice() method shall always return null.

11.4.2.8.2 TV Behaviour Control

The "TV Behaviour Control" in figure 31, "Format control in the presence of a JMF player", is the default video transformation behaviour for JMF players which have none other set. Setting the VideoTransformation to DFC_PLATFORM shall set the video format control to TV behaviour control mode.

NOTE 1: In this mode, any changes in the video input signal where "television behaviour control" requires changing the pixel aspect ratio of the final output signal (i.e. after the combination of video with graphics, subtitles and backgrounds) will also change the pixel aspect ratio of any graphics, subtitles and backgrounds combined into that one final output signal, assuming the JMF player has successfully reserved the HVideoDevice and can configure it appropriately.

NOTE 2: For an example of "TV behaviour control" in set-top boxes and integrated digital TVs, see E-Book.

NOTE 3: Where the JMF player is unable to change the configuration of the HVideoDevice, "TV behaviour control" takes into account the active HVideoConfiguration when determining the decoder format conversion to apply. This situation may occur if another application has reserved the HVideoDevice or if an application has set a conflicting configuration on another HScreenDevice of the HScreen.

11.4.2.8.3 Application Behaviour Control

When a GEM application is controlling video presentation through BackgroundVideoPresentationControl (for video in the background) or VideoPresentationControl and methods on java.awt.Component (for video in a component), the application is responsible for monitoring the incoming video format and changing the video presentation if desired.

Instances of VideoTransformation constructed using the constructor of that class shall not impact the configuration of the HVideoDevice. Instances of VideoTransformation returned by VideoFormatControl.getVideoTransformation shall have the same impact on the HVideoDevice as the equivalent conversion applied in "TV behaviour control mode". These may not be fully achievable if the JMF player does not have the HVideoDevice reserved or if an application has set a conflicting configuration on another HScreenDevice of the HScreen.

11.4.2.8.4 Dynamic Behaviour

A JMF player shall attempt to reserve the HVideoDevice instance on entry to the Prefetched state from any state except the Started state. A JMF player in the prefetched or started states which does not have the HVideoDevice reserved shall attempt to reserve it only when the application which created the JMF player calls the setVideoTransformation method of the control org.dvb.media.BackgroundVideoPresentationControl. A JMF player leaving the Started or Prefetched state for the Realized state shall release the HVideoDevice if it has it reserved.

A JMF player which has the HVideoDevice reserved and then loses the right to control that device shall post an instance of the event org.davic.media.ResourceWithdrawnEvent to all registered listeners for

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 145 of 861

ControllerEvents on that Player instance. If the right to control that device is subsequently recovered when a attempt to reserve the device (as defined above) succeeds, then an org.davic.media.ResourceReturnedEvent shall be posted.

11.4.2.8.5 Resource Management Details

Inter-operable applications shall not call any of the methods on the instance of ResourceClient returned by HVideoDevice.getClient when a JMF player has reserved that HVideoDevice instance.

Inter-operable applications are allowed to call HVideoDevice.releaseDevice or HVideoDevice.setConfiguration directly. GEM terminals shall continue to present video on a best effort basis within the constraints imposed by the new video configuration.

11.4.2.9 Integration with providers

The following shall be supported to enable the streamed media API to be used in combination with the provider API (see clause 11.7.9, "Provider API"). Since service providers (clause 11.7.9) are optional in GEM, this clause is not required by GEM terminal specifications.

- An implementation specific DataSource (referred to as ProviderDataSource) that is aware of which SelectionProvider objects are in scope for a particular application. This DataSource shall check if the protocol of the MediaLocator is one for which a service provider in scope of that application has registered as a handler. The getContentType method of this DataSource is assumed to have a hard-coded return value of "multipart.dvb.service" (according to syntax specified by ContentDescriptor.mimeTypeToPackageName).

NOTE: The preceding requirement is that a class with that behaviour be present. It need not have the name ProviderDataSource in implementations.

- An implementation specific Player that can use instances of ProviderDataSource as input, i.e. which will not throw a IncompatibleSourceException when a ProviderDataSource is passed to its setSource method. For the purposes of the "algorithm for creating a Player from a MediaLocator", this is assumed to be listed in the content package-prefix-list as supporting the media-content-type-name "multipart.dvb.service", i.e. the content type supported by ProviderDataSource.

11.4.2.10 Additional and modified semantics for IPTV

For services and content items available through RTSP, calls to the method Player.setRate shall result in a request to change the scale value as defined in clause 12.34 of RFC 2326 [83]. The return value of the method shall be the actual scale value chosen by the server.

NOTE: There is no standardized mechanism for obtaining the set of available rates from a server. GEM applications wanting that information need to obtain it from the server themselves.

11.4.2.11 Time-setting operations for OTT

In an OTT target, when the methods involved in changing the current media time, i.e. javax.media.Player.setMediaTime() and org.davic.media.MediaTimePositionControl.setMediaTimePosition() are invoked, implementations will achieve the result expected by applications, transparently to the applications themselves. The way the result is actually achieved is implementation-dependant and may also depend upon the server (e.g. lack of HTTP Range header support) and/or the content (fixed vs adaptive rate).

After the player has successfully retrieved the part of the file containing the indicated media-time and the player has re-entered the started state a javax.media.MediaTimeSetEvent will be generated.

Implementations will adjust the time values passed to the above mentioned methods to the most appropriate time point available in the container.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.5 Data access APIs

11.5.1 Broadcast Transport Protocol Access API

The broadcast transport protocol access API is defined in P "(normative): Broadcast Transport Protocol Access".

Relative file names used to access objects in the application filesystem shall be taken as being relative to the base directory indicated in the Application Description defined in clause 10.5, "DVB-J specific Application Description".

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

The caching rules specified in clause B.5, "Caching" shall be evaluated at the time when the information is loaded using `DSMCCObject.synchronousLoad()` or `DSMCCObject.asyncronousLoad()` or is loaded implicitly in response to other actions as defined in clause 11.5.1.1, "Constraints on the java.io.File methods for broadcast carousels". After the DVB-J object has been loaded, any possible changes in the object carousel to the content the object represents are not visible to the application when it accesses the content using this DVB-J object instance.

11.5.1.1 Constraints on the java.io.File methods for broadcast carousels

The application shall be able to use the standard `java.io.File` class for access to broadcast carousels (e.g. a carousel unaware application). In this case, the following definitions shall apply:

- The constructor of `File` only creates an instance of the abstract pathname and shall not cause synchronous access to the broadcast carousel that would block the thread.
- After the constructor of the DVB-J `File` object has been run, the directory entry information relating to this object may be in an unloaded state. The information relating to this object (e.g. its length) comes from the parent directory which is not required to have been loaded at this point by the constructor. However, this information may be available if the implementation has the information in cache.
- If the directory entry information for a DVB-J `File` object is in the unloaded state, then this information shall be synchronously loaded when any of the following methods are called on the object: `can_Read()`, `exists()`, `isDirectory()`, `isFile()`. If the loading fails, all these methods shall return false.
- If the content is in unloaded state, it shall be synchronously loaded when the `list()` method is called for a directory. If this implicit load should result in a service transfer, it shall not be done implicitly and the list() shall return an empty list.
- The method `lastModified()` returns a value as specified in the GEM terminal specification. GEM terminal specifications shall define signalling for this value to indicate at least 256 distinct values.

NOTE 1: Even if the "carousel" functional equivalent is adopted by a GEM terminal specification, the above applies. Such specifications have to define the binding for `lastModified()`, even if this binding is the same as MHP's.

NOTE 2: GEM terminal specifications may, of course, define a binding that provides for a larger set of values than the minimum required for the method `lastModified()`.

- Any version changes in a file after the constructor for an `FileInputStream`, `FileReader` or `RandomAccessFile` is called or an `InputStream` referring to a file is obtained through other means (e.g. via `javax.microedition.io.InputConnection`) will not be visible in the data read from that instance.
- The creation of a `FileInputStream`, `FileReader` or `RandomAccessFile` shall throw `FileNotFoundException` if the referenced object is not a file.
- Failure of a signed file to be authenticated shall be reported as defined in clause 12.6.1, "General principles".
- Failure of a signed directory to be authenticated shall be reported by the list methods returning null when called on a file object representing the directory whose authentication failed.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- The value returned by `java.io.File.length()` may not be accurate as it returns the value from the directory information rather than the actual size of the file.

There are no guarantees that the most recent version of a file will be returned unless the network signalling specifies that the file concerned requires transparent access.

NOTE 3: Although jar files may be accessed, using them extensively is not recommended within object carousels. Accessing files contained within a jar file that is contained within an object carousel is likely to be slower than accessing the same file when the jar file is unpacked into an object carousel.

11.5.1.2 Methods dealing with write access

The `java.io.File` class also contains methods that assume write access to the file system. Due to its broadcast nature, the receiver naturally does not have write access to the carousel.

NOTE: A broadcast carousel is not a read-only file system (which has the property of not changing). The carousel content can certainly be written and modified, but only by the service provider - not the receiver. Therefore, the situation is equivalent to a Unix file system where the user has only read permissions, but not write permissions or ownership of the files.

The following `java.io.File` methods deal with write access to directories: `canWrite()`, `mkdir()`, `mkdirs()`, `renameTo()`.

For abstract pathname entries in the broadcast carousel, the following behaviour shall apply:

- `canWrite()` returns `false` to indicate that the file can not be written to.
- `mkdir()`, `mkdirs()` and `renameTo()` return `false` to indicate that the request failed.

11.5.1.3 Behaviour following loss of a broadcast carousel

When a broadcast carousel is lost to an application as described in clause 6.2.5.3, "Loss of carousel behaviour", the following failure modes shall be followed for data which is unavailable.

- Attempting to load data from a file using a content format specific API shall fail as if the file itself never existed. This shall be done according to the specific API concerned.
- The classloader created by the platform when the application was first launched never attempts to automatically recover following loss of its initial broadcast file system.
- Attempting to load a class which is unavailable shall fail as if the class was never present.
- After the constructor for an `InputStream` has been successfully called, the data for that `InputStream` shall be maintained by the platform until the `InputStream` is closed. This is the same behaviour as specified for `InputStream` and file version changes in clause 11.5.1.1, "Constraints on the java.io.File methods for broadcast carousels".
- Attempting to create a `FileInputStream` for a file whose data is unavailable shall fail with a `FileNotFoundException`.
- Attempting to create a `RandomAccessFile` for a file whose data is unavailable shall fail with an `IOException`.
- Attempting to use a `FileDescriptor` of a `FileInputStream` whose data is unavailable shall result in a FileInputStream or InputStreamReader where all methods shall throw an `IOException`.
- Attempting to call methods on a `File` object where the filesystem information needed for the method call (see clause 11.5.1.1, "Constraints on the java.io.File methods for broadcast carousels") is unavailable shall fail in the same way as if the `File` itself never existed.
- Operations ongoing at the time of loss of broadcast carousel shall be terminated. Blocked Java I/O operations shall throw `InterruptedIOException`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.5.2 Support for Multicast IP over the Broadcast Channel

Where support for carriage of multicast IP over the broadcast channel (see clause 6.2.6) is included, the following shall apply:

a) In `java.net.MulticastSocket`, the `send` method shall throw an `IOException` when called on sockets bound to unidirectional sources of IP multicast data.

b) In `java.net.DatagramSocket`, the `send` method shall throw an `IOException` when called on sockets bound to unidirectional sources of IP multicast data.

c) The class `org.dvb.net.DatagramSocketBufferControl` shall be supported see annex Q, "Datagram socket buffer control".

d) Behaviour if unsupported:

   When the application tries to `joinGroup()` on a Multicast address, ProtocolException shall be thrown to indicate failure on joining the group.

11.5.3 Support for IP over the Return Channel

NOTE 1: Where support for IP over the return channel is not included, APIs in the `java.net` package will fail as defined in the specification of this API. Where support for IP over the return channel is supported, platforms not implementing specific optional return channel protocols will fail as defined in the specification of this API.

On devices whose return channel can be connected or disconnected, connecting a `java.net.Socket` or a `java.net.URLConnection` to a host addressed via the return channel shall automatically setup a connection to the default connection target subject to the application having return channel permission for the default ISP and the return channel either being not connected or being connected to a different target but with the return channel resource reserved by another application with a lower priority. Such connections shall be automatically disconnected after a period of inactivity on that connection which it should be possible to define using the navigator. This period shall be returned to applications in the `"dvb.returnchannel.timeout"` system property encoded in seconds as a decimal number.

See clause 11.10, "Java permissions".

The class `org.dvb.net.DatagramSocketBufferControl` shall be supported see annex Q, "Datagram socket buffer control".

NOTE 2: Support multicast of IP over the return channel is not required by clause 6.3, "Interaction channel protocols". On GEM terminals not supporting this, the methods concerned will fail using one of their defined failure modes.

11.5.4 MPEG-2 Section Filter API

The MPEG-2 section filter API is defined in annex E of DAVIC 1.4.1p9 [17].

11.5.5 Mid-Level Communications API

See annex R, "DVB-J return channel connection management API".

On devices whose return channel can be connected or disconnected, when use of `java.net.Socket` or a `java.net.URLConnection` automatically sets up a connection, this shall be reported through this API to any applications listening for it.

Implementations shall automatically drop connections established using this API after the same period of inactivity as defined for automatically setup connections in clause 11.5.3, "Support for IP over the Return Channel".

If an application which has reserved a `ConnectionRCInterface` and established a connection then releases the resource without disconnecting, the behaviour of the GEM terminal is implementation dependent.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

If there are open sockets using that connection, the GEM terminal should keep the connection established until the time-out period defined in clause 11.5.3, "Support for IP over the Return Channel" for automatically setup connections.

NOTE: On GEM terminals not including a return channel, this API is present but will report that no `RCInterface` instances are visible to any application.

11.5.6 Persistent Storage API

- The API to persistent storage shall be the `javax.microedition.io` package, the `java.io` package and the extensions to it found in the `org.dvb.io.persistent` defined in annex K, "DVB-J persistent storage API".
- The "`dvb.persistent.root`" property which can be obtained from `java.lang.System.getProperty()` identifies the directory at the root of the file name space used for persistent storage. This value shall be the same for all applications.
- Accessing any files or directories in the parent directory of this root or directories above that shall throw a SecurityException as defined in the specification for the `java.io` package.
- Signed applications which are authenticated and which are granted the right to access persistent storage have the following privileges:
  - Read only access to the root directory (defined above).
  - Automatic read and write access to an "organization" sub-directory named <`organisation_id`>.
  - Automatic read and write access to an "application" sub-directory named <`organisation_id`> + <`file.separator`> + <`application_id`>.
- When the permission request file for an application requests access to persistent storage and this is granted, the GEM terminal shall be responsible for creating the necessary directories in which the application is allowed to read/write where these do not already exist. This shall be done shall be done no later than immediately prior to the first access (including existence checks such as `File.isDirectory()`) by that application instance to the directories concerned.

  For applications which are granted file access (see clause 12.6.2.7, "File Access"), the necessary directories are only its organization and application sub-directories. For applications which are granted a credential, (see clause 12.6.2.6, "Credentials"), the necessary directories are those defined by any filename elements in that credential up to but not including the first directory containing a wildcard.

EXAMPLE: A filename element such as:

`org0_id/appA_id/external/-`

creates

`org0_id/appA_id/external`

When the GEM terminal automatically creates an "application" sub-directory as part of the necessary directories defined above, it shall set the owner of the created sub-directory to the application whose name corresponds to that sub-directory and set read and write access to the application whose name corresponds to that sub-directory and no access for other applications. The GEM terminal shall set the owner of any additional automatically created necessary directories that reside below an "application" sub-directory to the application whose name corresponds to that "application" sub-directory and set read and write access to the application whose name corresponds to that "application" sub-directory and no access for other applications.

If at the time when a GEM terminal would create the "application" sub-directory (if it did not exist), that directory already exists but is owned by an application other than the one whose `application_id` is used as its name, the GEM terminal shall remove this directory and any contents of it and create an empty "application" sub-directory with the owner and access rights set as defined above.

- The owner of the "organization" directory shall be the platform and the directory itself shall always have organization read / write and world read access.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 150 of 861

- All files in persistent storage shall store the 48 bit application identifier of their creator as the "owner" of the file. Only the owner of the file is entitled to change the file attributes and file access rights. The owner of a file cannot be changed once a file is created.
- Applications may only create files or directories in directories to which they have write access.
- The existing semantics for the java.io package are respected.
- See clause 14.5, "Text encoding of application identifiers". The fields <organisation_id> and <application_id> are hexadecimal text encodings (without leading zeros) of the fields in the 48 bit application identifier of the application concerned as defined in clause 10.4.3, "Content of the Application Description".
- All access to persistent storage shall be consistent with the security mechanism defined in clause 12, "Security".

NOTE 1: As defined in clause 11.5.1, "Broadcast Transport Protocol Access API", relative paths cannot be used to access persistent storage in an inter-operable way.

NOTE 2: Unsigned applications have no access to persistent storage. See clause 12.6.2.7.1, "Unsigned applications".

The contents of a file placed in persistent storage shall persist at least until one of the following conditions occur:

a) There is not enough free persistent storage available to satisfy the persistent storage needs of a running application.

b) The file stored in persistent storage expires.

c) The file is cleared (i.e. deleted) by the creating application or an application with appropriate permissions.

d) The file is cleared as a result of an explicit request by the end user.

e) The persistent storage used by GEM applications exceeds 75 % of the value in table 88, "Minimum requirements for other resources for conformance purposes".

NOTE 3: The behaviour of the GEM terminal when any of the above conditions occur is implementation specific except as follows:

Each organisation that has a GEM organisation_id may have one special file in persistent storage. This file is identified by having the name "prefs.bin" and being located in the sub-directory immediately below the root of the persistent file namespace whose name is the organisation_id of the organisation concerned. As long as the special file has a size less than or equal to 2 K bytes, the following behaviour applies:

a) GEM terminals do not automatically delete these special files except when persistent storage is full and all files in persistent storage are these special files.

b) GEM terminals that offer the end-user a user interface to manage the contents of persistent storage should present these special files such that they are less likely to be deleted by the end-user than any files that are not special.

The terminal shall give priority to the retention of higher priority files over lower priority files within the scope of a specific(organization ID, application ID) pair. See org.dvb.io.persistent.FileAttributes for priorities.

The details regarding the release of cleared (i.e.deleted) or expired files are implementation dependent.

The GEM terminal shall have a means to clear at least the quantity of persistent storage defined in table 88, "Minimum requirements for other resources for conformance purposes". This mechanism shall be usable in conformance tests, and is not subject to the rules above.

Some GEM terminals may allow a file to be opened for writing by only one FileOutputStream (or other file-writing object) at a time. In such situations the constructors in FileOutputStream, RandomAccessFile and FileWriter will fail if the file involved is already open. Concurrently running applications writing to the same file(s) in persistent storage cannot rely on this mechanism and will need to co-ordinate access to persistent storage themselves.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 151 of 861

11.5.7 File Storage Device Access

11.5.7.1 Basic Specification

The API for access to file storage devices is the `org.ocap.storage` package defined in clause 13.3.7.5 and annex V of OCAP [5].

11.5.7.2 DVB specific modifications

The following modifications apply to the `org.ocap.storage` package in the context of the present document:

- The requirement that "Persistent storage device contained within or attached to a Host device shall be represented by a StorageProxy" does not apply in the present document. Only detachable and removable devices need be represented in this way.
- The "storage" target for the class `MonitorAppPermission` is not defined in the context of the present document. Exceptions stated as being thrown for applications that do not have `MonitorAppPermission("storage")` shall always be thrown. Hence, the methods `StorageProxy.deleteVolume` and `StorageProxy.initalize` are not required to be implemented in GEM terminals. It is expected that the initialize feature would be provided by the resident navigator in GEM terminals supporting access to memory cards.
- GEM persistent file systems are not required to support the extended attributes defined in the class `org.ocap.storage.ExtendedFileAccessPermissions`. The method `org.dvb.io.persistent.FileAttributes.setFileAttributes` shall throw an IOException when called for a file whose file system does not support these extended attributes.

11.6 Service information and selection APIs

11.6.1 Signalling-specific service information API

There is no signalling specific SI API defined in GEM. GEM terminal specifications shall provide any needed service information API specific to their signalling.

11.6.2 Service selection API

The service selection API is defined by the `javax.tv.service.selection` package from Java TV [16] and the `org.dvb.service.selection` package defined in annex AK.

All instances of `ServiceContext` returned by the GEM terminal shall be instances of `DvbServiceContext`.

On the first occasion when a the JMF player is obtained for a specific service (either by ServiceContext.getServiceContentHandlers or HVideoDevice.getVideoController), any JMF players returned shall always be in the started state If they are presenting video, that video shall always be presenting on the background video device unless all of the following apply:

- the previous service presented in that service context contained video;
- that video was presented in a Component (i.e. the ServiceMediaHandler presenting it was a component based player);
- the component continues to be displayed after the completion of the selection of the current service in the service context.

If all of these conditions are met then the video of the new service shall be displayed in the same Component as the video of the previous service and the JMF player used shall be a component based player.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Applications shall re-acquire the list of service content handlers after a service selection completes. It is implementation dependent whether any previous handlers are re-used. JMF players which are not re-used are stopped and may be disposed by the platform. Where a JMF player is re-used, it shall be reset to a default condition as if it was not being reused. In particular the implementation shall cancel all listeners previously defined and reset previously defined characteristics such as audio language. If the video is presented in a background video device and no default video transformation is set for the service context then video scaling and positioning shall be reset.

The exception `javax.tv.services.selection.ServiceContextException` shall never be thrown by GEM terminals.

The default media player behaviour following service selection is defined under clause 11.4.2.3, "Default media player behaviour".

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

Implementations of this API are required to perform tuning as part of implementing the `ServiceContext.select` methods where the new service to be selected is part of a transport stream known to the GEM terminal but not currently tuned to.

All the GEM applications running within the same service context shall have the ability to obtain and modify the state of the service component handlers for all service components being presented in that `ServiceContext`. For service component handlers which are JMF players, modifying the state includes changes to the settings of JMF controls in addition to the state machine of the JMF player itself.

Applications not running in that service context (e.g. the GEM navigator) shall not be able to modify the state of service component handlers unless they first inform the applications in the service context of this. For service component handlers which are JMF players, this informing shall be done by sending a `org.davic.media.ResourceWithdrawnEvent` to all applications which have currently registered listeners for ControllerEvents on the JMF player whose state will be modified.

Once a `ResourceWithdrawnEvent` has been sent, any changes made by applications inside the service context to the state of service component handlers will be ignored by the GEM terminal. Methods to query state shall return the actual state and methods to change the state shall silently fail. It is implementation dependent whether events related to JMF continue to be sent to applications.

If the application outside the service context ceases to need to make changes to the state of service component handlers, the applications inside the service context shall return to having that right exclusively. The applications inside the service context shall be informed of the return of this right. For service component handlers which are JMF players, this informing shall be done by sending a `org.davic.media.ResourceReturnedEvent` to all applications which have currently registered listeners for `ControllerEvents` on the JMF player which is returned to exclusive control.

Following receipt of a `ResourceReturnedEvent`, the GEM applications running in the service context are responsible for returning the configuration of all service component handlers to what they require. The GEM terminal is not required to automatically restore any configuration of any service component handler to any former value. In the case of a service component handler which is a JMF Player, the configuration is the set of writeable properties accessed through the `Controls` of that `Player`.

NOTE 1: Example properties include video scaling, video positioning and audio language choice.

NOTE 2: One situation where this may occur is where the end-user of the GEM terminal brings up the UI of the navigator, does some operation and then exits the navigator, returning control back to the GEM applications.

NOTE 3: There is no relationship required between the generation of these events and the Xlet state model. GEM terminals may chose to put all GEM applications which are in the Active state into the Paused state when the bring up the UI of the navigator however there is no requirement for this.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A running DVB-J application shall be considered as a component of the service being presented in the `ServiceContext` in which the application is running. It shall have a `ServiceContentHandler` which is intentionally not defined by the present document except that it shall not be a `ServiceMediaHandler`. The results of calling the `getServiceContentLocators` method on this `ServiceContentHandler` are implementation dependent, and there is no requirement for these Locators to be accepted as input by any method in the present document. If the `ServiceContext` in which a DVB-J application is running is stopped, the DVB-J application shall be stopped like all other service components being presented in that service context. `ServiceContentHandlers` for DVB-J applications shall not be shared between DVB-J applications if more than one application is running.

NOTE 4: An Xlet is not a `javax.tv.service.navigation.ServiceComponent`, the elementary stream(s) carrying the object carousel carrying the Xlet are represented by at least one of these.

NOTE 5: Services with only DVB-J applications and no media components will have no `ServiceMediaHandlers` but only `ServiceContentHandlers` representing the running DVB-J applications.

NOTE 6: The present document intentionally does not define any circumstances under which `SelectionFailedEvent(MISSING_HANDLER)` is required to be generated.

NOTE 7: Stopping all the components of a service (both media and Xlets) results in a service with no components being presented, the same as if a service with no components had been initially selected.

The following stream types (from `javax.tv.service.navigation.StreamType`) shall be selectable: AUDIO, VIDEO, SUBTITLES. Service components of these types shall have a `ServiceContentHandler` which may be shared between multiple service components of the same or different stream types. It is implementation-dependent whether other stream types are selectable, but failure when selecting a stream type that is not selectable is handled as defined in the present document. The effect of successfully selecting a stream type that is not in the preceding list is implementation-dependent.

If the selected service is a DVB NVOD reference service, GEM terminals shall make an implementation dependent choice from those NVOD time shifted service associated with that reference service and try to select the chosen time shifted service. The algorithm for the implementation dependent choice shall be automatic and not involve the end-user. Any failure shall be reported as specified through this API.

NOTE 8: Applications which wish to chose a specific time shifted service need to explicitly specify that service and not rely on the choice of the receiver.

This API shall support the following features on those GEM terminals which support the feature concerned.

- Internet clients (see clauses 9.7, "Lifecycle of internet access applications" and 11.14.1, "Internet client control APIs").
- Stored services (see clauses 9.6.2, "Stored services" and 11.12.2.2, "Modified behaviour of GEM 1.0 APIs").
- Services signalled over the interaction channel (see clauses 9.6.1, "Applications loaded from the interaction channel" and 11.11.12, "Support for the HTTP Protocol in DVB-J").

This API shall support the following features:

- Services whose locator is an instance of the `org.dvb.locator.FrequencyLocator` class.
- Service instances representing PSI-only services.

Service contexts shall not have any hard coded restrictions on the types of service that can be selected in them. For example, a service context which is at one point presenting a broadcast service may not fail to have a stored service selected in it for no other reason than the service to be selected being a stored service.

In the method `ServiceContext.select(Locator[])`:

- The language applying to Xlets shall apply to all GEM applications regardless of type.
- Xlets can be listed using the application locator, see MHP [1], clause 14.1.7, "Application locator".
- Any listed locators for applications not signalled as AUTOSTART or PRESENT shall be ignored.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 154 of 861

Subclasses of `javax.tv.service.selection.PresentationChangedEvent` shall be associated in time with the generation of a `MediaPresentedEvent` and not the JMF state change event which may have happened some seconds previously.

Re-selecting the currently selected service in a service context shall be as defined in the `javax.tv.service.selection` package with the additional clarification that running applications whose `service_bound_flag` is set to 1 shall not be killed and re-started if the Xlet is supposed to be running after the selection operation completes.

Completion of a service selection operation (whether reported via `NormalContentEvent`, `AlternativeContentEvent` or `SelectionFailedEvent`) shall be reported once all `ServiceMediaHandler` instances are in the started state, without waiting for any xlets to enter the active state.

11.6.3 Tuning API

GEM terminal specifications for packaged media targets are not required to support a tuner. The definitions and requirements of this clause apply only to GEM terminal specifications that support a tuner.

11.6.3.1 Generic description

The tuning API is defined in annex H of DAVIC 1.4.1p9 [17] apart from section H.4, (the `Locator` and `DvbLocator` classes) which are found in clause 11.7.6, "Content referencing". The reference to the `DvbLocator` class does not apply to GEM terminal specifications that do not include the functional equivalent named "Content Referencing" as defined in clause 15.6, "Functional equivalents".

The following methods in this package shall throw `java.lang.SecurityException` where and only where the application does not have a `org.dvb.net.tuning.TunerPermission`:

- `NetworkInterfaceController.reserve`
- `NetworkInterfaceController.reserveFor`

The class `org.davic.net.tuning.dvb.DvbNetworkInterfaceSIUtil` is not required.

See also clause 11.8.3, "Additional permissions classes".

Tuning automatically performed by other APIs in GEM shall be reported through the tuning API to any applications listening for it.

11.6.3.2 Tuning in IPTV

In IPTV, the tuning API shall represent control over the input to the MPEG-2 demultiplexer. Hence tuning to a transport stream means connecting that transport stream to this input.

11.6.3.3 Tuning in OTT

In OTT, the tuning API shall represent control over the input to the stream decoder / demultiplexer. Hence tuning to a transport stream means connecting that transport stream to this input.

11.6.4 Conditional access API

The present document does not require a conditional access subsystem, nor does it place requirements on one if it is present.

GEM terminal specifications may define any needed conditional access API.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.6.5 Protocol independent SI API

11.6.5.1 Generic description

The protocol independent SI API is defined by the following packages from Java TV [16]:

- `javax.tv.service`
- `javax.tv.service.guide`
- `javax.tv.service.navigation`
- `javax.tv.service.transport`

The mapping of this onto the DVB-SI protocol is specified in annex O, "Integration of the JavaTV SI API". The mapping of this onto services whose service information is obtained through a provider is defined in clause 11.7.9, "Provider API".

The mapping of the protocol independent SI API onto the underlying SI protocol is not defined in GEM. Thus, the reference to annex O does not apply for GEM terminal specifications that do not include the functional equivalent named "SI" as defined in clause 15.6, "Functional equivalents". However, GEM terminal specifications shall provide a mapping of the protocol independent SI API onto their SI signalling, as specified in annex O.

Cancellation shall fail if the request is no longer pending. It is implementation dependent if cancellation fails under any other circumstances. Implementations are not required to support cancellation of requests between when the requested SI data has arrived in the device and when execution of the `notifySuccess` method starts.

The interface `javax.tv.service.ServiceMinorNumber` is not required to be implemented by any object defined in GEM.

This API shall support the following features on those GEM terminals which support the feature concerned.

- Internet clients (see clauses 9.7, "Lifecycle of internet access applications" and 11.14.1, "Internet client control APIs").
- Stored services (see clauses 9.6.2, "Stored services" and 11.12.2.2, "Modified behaviour of GEM 1.0 APIs").

The method javax.tv.service.SIManager.getService(Locator) shall, if passed an instance of org.dvb.locator.FrequencyLocator or a "dvb:" locator referencing a PSI-only service, return an instance of javax.tv.service.Service representing the service concerned. The service returned shall have the following characteristics:

- the `getName` method shall return a string of length zero;
- the `getServiceType` method shall return UNKNOWN;
- the `retrieveDetails` method shall report `notifyFailure` with `SIRequestFailureType(DATA_UNAVAILABLE)`.

Such services shall never be returned to a GEM application instance from a call to `SIManager.filterServices`.

A service of these characteristics shall be returned even if the service referenced by the `FrequencyLocator` is one in the service list of the receiver for which the name, type and details are known.

For the purposes of `SIManager.getService(Locator)` throwing an `InvalidLocatorException`, the `FrequencyLocator` shall be considered valid if and only if the GEM terminal has a `NetworkInterface` of the same type (cable/satellite/terrestrial) as the descriptor used to construct the Locator.

11.6.5.2 Transport independent and dependent services

The following clarifications shall apply to Java TV as used in GEM.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- All instances of Service that are not instances of a standardised sub-type of Service (e.g. AbstractService, StoredApplicationService or InternetClientService) shall be instances of either TransportIndependentService or TransportDependentService.
- Instances of TransportDependentService shall not appear in the list of service returned by SIManager.filterServices, even when null is passed to that method.

NOTE: As a consequence, DeliverySystemTypeFilter cannot return instances of TransportDependentService.

- SIManager.getService shall return an instance of TransportDependentService when passed a transport dependent locator.
- ServiceDetails.getService shall not return a TransportDependentService
- References to "broadcast service information" shall be interpreted to include service information in the IP network as defined in clause 6.5.2, "Service information and metadata protocols" of the present document.

11.6.5.3 Modified Semantics of Existing APIs

Clause T.2.2.16 of OCAP [5] shall be supported for instances of javax.tv.service.navigation.ServiceDetails corresponding to abstract services.

11.6.5.4 Extensions

This is the org.dvb.service and org.dvb.service.navigation packages.

11.6.6 Service discovery and selection for IPTV

This is the org.dvb.service.sds package.

The org.dvb.service.DvbSIManager class shall implement org.dvb.service.sds.SDSRecordAccess.

The `org.dvb.service.sds` package is defined in annex AW.

11.6.7 Integration between protocol independent SI API and TV-Anytime

This is the org.dvb.tvanytime.javatv package.

The org.dvb.service.DvbSIManager class shall implement the org.dvb.tvanytime.javatv.CRIDAccess interface.

Instances of javax.tv.service.guide.ProgramEvent whose data was obtained via the TV-Anytime protocols used in TS 102 539 [81] shall implement org.dvb.tvanytime.javatv.TVAProgramEvent.

The `org.dvb.tvanytime.javatv` package is defined in annex AY.

11.7 Common infrastructure APIs

11.7.1 APIs to support DVB-J application lifecycle

This API is formed of the Java classes and interfaces found in the `javax.tv.xlet package` specified in Java TV [16]. In addition, the package `javax.microedition.xlet` contains similar classes. In either case, the `XletContext` passed to the `Xlet` shall implement both `javax.microedition.xlet.XletContext` and `javax.tv.xlet.XletContext`.

GEM terminals shall determine which Xlet interface is implemented by the initial class of an application, and use the corresponding application management API. In the case where the initial class of an GEM application implements both `javax.tv.xlet.Xlet` and `javax.microedition.xlet.Xlet`, the GEM terminal shall use the `javax.tv.xlet` application management API.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 157 of 861

11.7.1.1 Xlet properties

The following named Xlet properties shall be supported:

- `dvb.org.id.`
- `dvb.app.id.`
- `dvb.caller.parameters.`

They can be retrieved from an Xlet's XletContext by calling `getXletProperty` with the string name defined above.

All keys for `XletContext.getXletProperty` beginning `'dvb.'` are reserved for use in DVB project specifications.

The array of strings returned by `XletContext.getXletProperty(XletContext.ARGS)` shall be the array of strings defined in the DVB-J application descriptor (see clause 10.5.2) in the same order as specified in the signalling. Each entry in the loop of that descriptor shall be presented as one string in the array returned by this method. Zero-length strings in the signalling shall be represented as the empty string.

The "dvb.caller.parameters" XletProperty contains the array of Strings that can be passed to applications started via AppProxy. If this application was started by the system or by another application without passing parameters via DVBJProxy (using either the start or init methods), an empty array is returned as the value of this XletProperty.

The "dvb.org.id" and "dvb.app.id" Xlet properties shall be initialised from the `application_identifier` of the application.

Table 31: Property name / type / encoding mapping

Property Name	getXletProperty return type
dvb.app.id	String encoded as in clause 14.5, "Text encoding of application identifiers".
dvb.org.id	String encoded as in clause 14.5, "Text encoding of application identifiers".
dvb.caller.parameters	String[], no information to be indicated with an array of length zero.
dvb.installer.parameters	getXletProperty return type String[] (Not required in GEM).

11.7.1.2 Actions for DVB-J applications to perform in their destroy method

Xlets shall perform at least the following in their `Xlet.destroyXlet` method and before calling `XletContext.notifyDestroyed`:

- Cause any threads that they have created to exit voluntarily.
- Stop, deallocate and close any JMF players that they have created.
- Stop and destroy any Java TV service selection ServiceContext objects that they created.
- Release any other scarce resources that they created, e.g. `NetworkInterfaceControllers` if they do any tuning.
- Flush any images using the `Image.flush()` method.
- Xlets shall not cause any unnecessary delay in their `Xlet.destroyXlet` method.
- De-register any event listeners.

If applications do not release these resources then the platform may do it for them.

Regardless of whether an application releases resources or whether the platform does it, the appropriate notifications shall be sent to all other GEM applications which are registered to receive these. For resources which are managed by an API using the `org.davic.resources` package, these notifications are specified by that API. For resources which are related to JMF players which are `ServiceMediaHanders`, if the GEM terminal implementation changes the configuration of these resources (e.g. to tidy-up), this shall be notified as defined in clause 11.6.2, "Service selection API" for "Applications not running in that service context".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

NOTE: Applications have to allow for tuning happening in parallel with the execution of their `destroyXlet` method. Hence implementations of that method should not rely on being able to load classes while executing as the carousel which the application was using may not be available.

11.7.2 Application discovery and launching APIs

This API is formed of the `org.dvb.application` package defined in annex S, "Application listing and launching".

In GEM terminal specifications that include the functional equivalent named "Carousel"defined in clause 15.6, "Functional equivalents", full support of all properties defined for use with the method `AppAttributes.getProperty` in table 32 is required. In GEM terminal specifications that do not include this functional equivalent, support of these properties is optional.

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

Table 32: Application attribute properties

Property name (see note)	Return
dvb.j.location.base	Returns String containing `base_directory_bytes` from DVB-J application location descriptor.
dvb.j.location.cpath.extension	Returns String[] derived from `classpath_extension_bytes` of DVB-J application location descriptor with each array entry corresponding to a pathname entry as defined for `classpath_extension_bytes`.
dvb.transport.oc.component.tag	Returns Integer containing the `component_tag` from the selector bytes of the transport protocol descriptor. If more than one transport protocol descriptor is in the AIT for a remote application then it is implementation dependent which of them is returned.
dvb.transport.oc.component.tag	Returns Integer containing the `component_tag` from the selector bytes of a transport protocol descriptor with `protocol_id` 0x0001. If more than one transport protocol descriptor with `protocol_id` 0x0001 is in the AIT for a remote application, it is implementation dependent which of them is returned.
dvb.ait.descriptors	See clause 11.7.8, "Plug-in APIs".
dvb.transport.oc.locator	An `org.davic.net.Locator` that identifies the object carousel. This shall be a Locator that is suitable for the `ServiceDomain.attach(Locator)` method. This shall be the Locator for the component specified by the "`dvb.transport.oc.component.tag`" property, in the service identified by `getServiceLocator()`.
NOTE: Property names beginning "`dvb.`" are reserved for future use.	

Where a property is specified as either containing a value from a particular descriptor or being derived from a particular descriptor, the property shall not be returned for applications for which that descriptor is not signalled. Where a property is specified as containing a value from a transport protocol descriptor with a particular `protocol_id`, the property shall not be returned for applications for which a transport protocol descriptor with that `protocol_id` is not signalled.

NOTE: The three properties `dvb.j.location.base`, `dvb.j.location.cpath.extension` and `dvb.transport.oc.component.tag`, defined above, are specific to transport via an MHP object carousel. They may not be available if other transport is used in other specifications, e.g. in a GEM terminal specification or a future version of the present document.

Table 33 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.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 33: Information source for methods on AppAttributes

Method	Information source
getName()	One of the names that can be found in the application_name of the Application description.
getName(String ISO639code)	A name of the application_name of the Application description corresponding to the specified language, if available.
getNames()	All of the names for the application which can be found in the application_name of the Application description and their ISO 639 [84] language code.
getProfiles()	The set of profiles indicated in the application_profile of the Application description.
getPriority()	The value indicated for the application_priority of the Application description.
getVersions(String profile)	The values version.major, version.minor and version.micro for the specified profile from the Application description.
getIsServiceBound()	True if the `service_bound_flag` of the Application description indicates true. Otherwise false.
isStartable()	There is no information source for this method, the return value is derived as specified in the method description. For the purpose of the method description, remote applications are as specified in the GEM terminal specification, if they are supported.
getIdentifier()	The `organisation_id` and application_id of the Application description.
getServiceLocator()	If remote applications are supported, the locator for a remote application shall encapsulate the values found in the appropriate signalling in the terminal specification.
getLocator()	The `application_icon_locator` of the Application description.
getIconFlags()	The `application_icon_flags` of the Application description.

11.7.3 Inter-application communication API

This API is formed of the packages `java.rmi` and `javax.microedition.xlet.ixc` as specified in PBP 1.1 [4] plus the package `org.dvb.io.ixc`.

Two named xlet properties are introduced: dvb.org.id and dvb.app.id. They can be retrieved from an xlet's XletContext by calling `getXletProperty("dvb.org.id")` and `getXletProperty("dvb.app.id")`, respectively.

The package `org.dvb.io.ixc` provides an access mechanism for the registry for registering and looking up remote objects for inter-xlet communication. This mechanism implicitly provides the organisation ID and application ID of an object when exported, but requires the caller to provide these values when importing an object. The `org.dvb.io.ixc` registry shall be mapped to the `javax.microedition.xlet.ixc` registry as follows.

Table 34: ixc registry name mapping

Type of access	GEM Registry Name	J2ME Registry Name
GLOBAL scope	name	dvb:/signed/organisation_id/application_id/name
GLOBAL scope	name	dvb:/unsigned/organisation_id/application_id/name
GLOBAL scope	/organisation_id/application_id/name	dvb:/signed/organisation_id/application_id/name
GLOBAL scope	/organisation_id/application_id/name	dvb:/unsigned/organisation_id/application_id/name
SERVICE scope	name	dvb:/service/service_context_id/signed/organisation_id/application_id/name
SERVICE scope	name	dvb:/service/service_context_id/unsigned/organisation_id/application_id/name
SERVICE scope	/organisation_id/application_id/name	dvb:/service/service_context_id/signed/organisation_id/application_id/name
SERVICE scope	/organisation_id/application_id/name	dvb:/service/service_context_id/unsigned/organisation_id/application_id/name

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

organisation_id and application_id refer to the organisation_id and application_id of the calling Xlet (see clause 10.4.3), and shall be formatted as specified in org.dvb.io.ixc.IxcRegsitry.lookup(). In the above table, service_context_id is a unique platform generated opaque name that identifies the service context for the purpose of SERVICE scope inter-xlet communication. The same name shall be used by all applications running in that service context.

The javadoc for the org.dvb.io.ixc package is provided in annex Y, "Inter-application and Inter-Xlet communication API". An example is provided in MHP [1], clause W.2 "Example of exporting an object for inter-application communication".

NOTE: Due to the security policy in clauses 12.6.2.14.1 and 12.6.2.14.2, all xlets can also import and export objects in the PBP registry under the "dvb:/ixc/*" namespace. See MHP [1], clause W.6 for an example of how two xlets can securely verify each others' identities.

11.7.4 Basic MPEG concepts

NOTE: The API from DAVIC defined in this clause is made optional for packaged media targets in clause 15, "Detailed platform profile definitions" because it relates to transport streams.

This API is formed of the Java classes defined in annex G of DAVIC 1.4.1p9 [17]:

- ApplicationOrigin
- ElementaryStream
- Service
- TransportStream

Methods returning instances of elementary stream. service or transport stream shall return instances of the DVB specific subclass (DvbElementaryStream etc.), if the GEM terminal specification includes these classes.

11.7.5 Resource notification

The resource notification API is defined in annex F of DAVIC 1.4.1p9 [17].

- The notifyRelease() method shall be called for all ResourceClients where the corresponding resource has been removed without the application releasing it. The release() method shall be called for specific resources where time to cleanup is of practical use considering the specific resource in question. The only one of these in the present document is connection oriented return channels.
- The requestData parameter of the requestRelease method shall implement the java.rmi.Remote interface. In APIs using the ResourceProxy interface, where the method to reserve the resource has a parameter 'requestData', this parameter shall also implement the java.rmi.Remote interface or be null. Use of other values shall result in null being used when requestRelease() is called. Implementing the Remote interface does not force java.rmi to be used when the current owner of the resource is in the same application as code requesting ownership.
- The resourceProxy.getClient() method shall always return the ResourceClient provided to the platform by the application when that instance of a ResourceProxy was created or reserved. The ResourceClient for the current owner of a resource is only returned when this method is called on the ResourceProxy which currently holds the resource concerned.

This text clarifies the DAVIC javadoc with respect to multiple simultaneous applications. Interpretations of the DAVIC javadoc which are inconsistent with this are excluded.

11.7.6 Content referencing

This API is formed of the DAVIC Locator class and the javax.tv.locator package.

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 161 of 861

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

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

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

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

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

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

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

11.7.7 Common error reporting

This API is formed of the interface and exceptions defined in annex G of DAVIC 1.4.1p9 [17]:

- `NotAuthorizedInterface`
- `NotAuthorizedException`
- `ObjectUnavailableException`
- `ResourceException`
- `TuningException`

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

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

Support for the `ObjectUnavailableException` and `ResourceException` classes is not required for GEM terminal specifications that do not include the optional support for 11.7.4, "Basic MPEG concepts" in the entry for "common infrastructure" in table Error! Reference source not found. of clause 15, "Detailed platform profile definitions".

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

11.7.8 Plug-in APIs

The plug-in API is defined in annexes AF, "Plug-in APIs" and AE, "Inner Applications" of the present document.

NOTE: This is the package org.dvb.application.plugins.

GEM inter-operable plug-ins signalled with the delegated application descriptor (see clause 10.7.3) shall implement the `org.dvb.application.plugins.Plugin` interface. Failure to implement this interface shall result in the plug-in being ignored. Plug-ins which only execute "delegated" applications signalled in the AIT do not need to implement the `javax.tv.xlet.Xlet` interface as part of the initial entry point from the application manager to the plug-in. Plug-ins which can execute both AIT and natively signalled "delegated" applications shall implement both the Plug-in and the Xlet interfaces.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The semantics of the method `org.dvb.application.AppAttributes.getProperty()` are extended such that when it is called with the string "`dvb.ait.descriptors`", an array of bytes containing the `application_descriptors_loop` shall be returned.

Each running instance of a plug-in shall execute in a single logical VM instance as defined in clause 11.2.1, "Basic Considerations". Hence all delegated application Xlets returned by the initApplication method are all executed in the same logical VM instance. It is the responsibility of the plug-in to ensure the presence or absence of shared context between instances of delegated application Xlets in the same logical VM instance.

11.7.9 Provider API

The provider APIs are defined in annex AN.

NOTE: This is the package `org.dvb.spi`.

11.7.9.1 API framework

This is the `org.dvb.spi` and `org.dvb.spi.util` packages.

The following tables define the mapping that shall be used between the provider APIs and the corresponding API as called by GEM applications.

11.7.9.2 SelectionProvider

This is the `org.dvb.spi.selection` package.

Table 35: Mapping from javax.tv.service.Service to provider APIs

Method	Mapping
getLocator	ServiceReference.getLocator.
getName	ServiceDescription.getLongName.
getServiceType	ServiceDescription.getServiceType.
hasMultipleInstances	Calculated by the receiver, returns true if and only if there are multiple services with the same transport independent locator.
retrieveDetails	SelectionProvider.getServiceDescriptions, either when the Java TV method is called or previously if the results have been cached and are still valid.

Table 36: javax.tv.navigation.ServiceDetails

Service	Mapping
addServiceComponentChangeListener	None needed.
removeServiceComponentChangeListener	
retrieveComponents	Not defined in the present document.
getDeliverySystemType	ServiceDescription.getDeliverySystemType.
getLongName	ServiceDescription.getLongName.
getProgramSchedule	No mapping required if schedule descriptions are obtained via standard SI. If schedule descriptions are obtained from a Provider then the mapping is to SimpleSIProvider.getScheduleDescriptions, either when the Java TV method is called or previously if the results have been cached and are still valid.
getService	Indirect - implementation must maintain enough information to lookup the service for a ServiceDetails.
getServiceType	ServiceDescription.getServiceType.
retrieveServiceDescription	Same as for DVB-SI in GEM 1.0 - always fails.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.7.9.3 SI providers

This is the `org.dvb.spi.si.simple` and `org.dvb.spi.si.full` packages.

Table 37: javax.tv.guide.ProgramSchedule

Service	Mapping
addListener	None needed. Events are generated after calls to programDescriptionAvailable and scheduleDescriptionAvailable.
removeListener	
getServiceLocator	Indirect - implementation must maintain enough information to lookup the service for a ProgramSchedule.
retrieveCurrentProgramEvent	SimpleSIProvider.getProgramDescriptions, either when the Java TV method is called or previously if the results have been cached and are still valid.
retrieveFutureProgramEvent	
retrieveFutureProgramEvents	
retrieveNextProgramEvent	
retrieveProgramEvent	

Table 38: javax.tv.guide.ProgramEvent

Service	Mapping
getDuration	Derived from ProgramDescription.getStartTime and ProgramDescription.getEndTime.
getStartTime	ProgramDescription.getStartTime.
getEndTime	ProgramDescription.getEndTime.
getName	ProgramDescription.getName.
getRating	Not defined in the present document.
getService	Indirect - implementation must maintain enough information to lookup the service for a ProgramEvent.
retrieveComponents	Not supported.
retrieveDescription	ProgramDescription.getDescription.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.7.9.4 InteractionChannelTransportProvider

The following table describes the interaction of DSMCCObject with the ICT service provider implementation, including differences in behaviour from classical broadcast carousel objects.

Table 39

DSMCCObject Method	Service Provider Method	Comments
DSMCCObject(String) DSMCCObject(String, String) DSMCCObject(DSMCCObject, String)	InteractionChannelTransportProvider.createResourceTransportObject(String)	
isLoaded()	ResourceTransportObject.isLoaded()	
isStream()	-	Not applicable; always return false.
isStreamEvent()	-	Not applicable; always return false.
isObjectKindKnown()	-	Not applicable; always return false.
synchronousLoad()	ResourceTransportObject.synchronousLoad()	
asynchronousLoad()	ResourceTransportObject.asynchronousLoad()	
abort()	ResourceTransportObject.abort()	
prefetch(String, byte) prefetch(DSMCCObject, String, byte)	InteractionChannelTransportProvider.prefetch(String, byte)	
unload()	ResourceTransportObject.unload();	
getURL()	-	Return the file: URL describing the abstract pathname with respect to the relevant logical file system root.
addObjectChangeEventListener(ObjectChangeEventListener)	ResourceTransportObject.addUpdateListener(ResourceUpdateListener)	
removeObjectChangeEventListener(ObjectChangeEventListener)	ResourceTransportObject.removeUpdateListener(ResourceUpdateListener)	
loadDirectoryEntry(AsynchronousLoadingEventListener)	ResourceTransportObject.loadMetadata(AsyncResourceLoadListener)	
setRetrievalMode(int)	-	Not applicable; has no effect.
getSigners() getSigners(boolean)	-	It is expected that the service provider implementation will implement whatever signature verification scheme is appropriate, so these methods will have no effect. An array of length 0 will be returned.

11.7.10 Content referencing for IPTV

This is the `org.dvb.locator.ip` package.

The IPTV content referencing API are defined in annex AU.

11.7.11 TV-Anytime content referencing and metadata

These are the `org.dvb.tvanytime.metadata`, and `org.dvb.tvanytime.resolution` packages from TS 102 816 [82] together with the `org.dvb.tvanytime.metadata.ip` package as defined in the present document.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 165 of 861

The following shall apply to instances of `javax.tv.locator.Locator` that reference ProgramEvents whose metadata is only available via the TV-Anytime protocols used in TS 102 539 [81].

- If the Locator instance was obtained by explicitly or implicitly resolving a CRID then the implementation shall remember the CRID concerned. Attempts to retrieve the ProgramEvent instance using such a Locator shall result in an attempt to retrieve the metadata associated with the remembered CRID.
- If the external form of the Locator matches the optional ProgramURL element in a TV-Anytime <ScheduleEvent> or <BroadcastEvent> fragment available to the GEM terminal, the GEM terminal shall obtain the CRID from that fragment and then attempt to retrieve the metadata for that CRID.
- If the external form of the Locator includes time and duration and the GEM terminal has schedule information corresponding to that entire duration then it should retrieve and return the metadata for the CRID with the greatest overlap with the time and duration of the locator.
- Otherwise attempts to retrieve the ProgramEvent instance for the Locator instance shall fail with a SIRequestFailureType of DATA_UNAVAILABLE.

For avoidance of doubt, there is no requirement for implementation to attempt reverse resolution of Locators to CRIDS.

The `org.dvb.tvanytime.metadata.ip` package is defined in annex AX.

11.7.12 Content referencing for OTT

This is part of the `org.dvb.locator.ott` package.

The OTT content referencing API is defined in annex AU.

11.8 Security

11.8.1 Basic Security

The GEM platform shall always install a `SecurityManager` before starting any DVB-J applications.

In GEM terminals where the `mhp.smartcard.reader` property is known and has the value "SUPPORTED", GEM applications shall be able to use a Provider installed by a GEM application as defined in annex AJ, "Cryptographics service provider installation," in those methods in the `java.security` package which have a Provider as an input parameter.

11.8.2 APIs for return channel security

This API is defined in the J2ME Security (JSSE) Optional Package Specification v1.0 which is found in FP 1.1 [78]. It includes the package `javax.microedition.io`.

NOTE 1: JSSE is optional in PBP 1.1 [4].

Cipher suites listed in table 56 "Profile of cipher suites that implementations are required to support" shall have the name found in the "cipher suite" column of that table with "TLS_" replaced by "SSL_". If other cipher suites from TLS RFC 2246 [58] are supported, it is implementation dependent whether their names start with "TLS_" or "SSL_".

NOTE 2: This language might be interpreted as disallowing these cipher suites to also be reported with a name beginning with "`TLS_`". GEM terminal implementations are not subject to such a restriction; the method `getSupportedCipherSuites()` on the concerned classes may return cipher suite names beginning with "`TLS_`" in addition to the names starting with "`SSL_`" that are required to be returned.

For GEM terminal specifications, this is to be read as requiring that these methods return at least the following constants:

```
SSL_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

```
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_WITH_DES_CBC_SHA
```

NOTE 3: `SSL_NULL_WITH_NULL_NULL` is not required to be returned. This cipher suite is used internally in the TLS/SSL protocol negotiation. Support of this protocol does not require that it be made available to applications via the getSupportedCipherSuites methods.

NOTE 4: If GEM terminal specifications add cipher suites in addition to the ones above, it may be appropriate for them to require the more current names beginning with `TLS_`, rather than names beginning with `SSL_`.

Also see clause 12.10, "Security on the return channel".

In GEM terminals where the `mhp.smartcard.reader` property is known and has the value "SUPPORTED", MHP applications shall be able to use a Provider installed by a GEM application as defined in annex AJ, "Cryptographics service provider installation" in both KeyManagerFactory and TrustManagerFactory.

11.8.3 Additional permissions classes

See annex T, "Permissions".

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

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

For GEM terminal specifications that do not support the "Tuning API", in clause 11.6.3, listed in table Error! Reference source not found., the class `org.dvb.net.tuning.TunerPermission` is not required.

11.8.4 General Security Issues

Unless otherwise specified, sub-classes of `java.security.Permission` are not required to check the input parameters to their constructors. When a GEM platform constructs these, it is responsible for creating them with legal values. The behaviour of a GEM platform if an application creates such an instance with illegal values is intentionally not specified.

11.8.5 Cryptographic API

This API is defined in the J2ME Security (JCE) Optional Package Specification v1.0 part of FP 1.1 [78].

NOTE: The JCE optional package is optional in PBP 1.1 [4].

In GEM terminals where the mhp.smartcard.reader property is known and has the value "SUPPORTED", GEM applications shall be able to use a Provider installed by a GEM application as defined in annex AJ, "Cryptographics service provider installation".

GEM terminals shall include a default provider for the `javax.crypto.Cipher` class as follows:

- In all GEM terminals, the RSA algorithm shall be supported.
- In GEM terminals implementing clause 12.10.2, "TLS cipher suites", the algorithms listed in that clause shall additionally be supported.

11.8.6 DVB Extensions for Cryptography

11.8.6.1 Introduction (informative)

"Java2 Standard Edition" includes APIs providing the ability to use PKCS11 tokens (such as smart cards) in Java applications. These APIs simplify the way in which Java applications can handle removable smart cards. They are also needed to login into PKCS11 token for non-key related operations such as encryption and decryption.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

GEM is based on J2ME Personal Basis Profile (PBP 1.1 [4]) which does not provide these new APIs. For this reason, the following new packages are defined in the present document, see annex AI, "DVB Extensions for cryptography".

- `org.dvb.security`
- `org.dvb.auth.callback`
- `org.dvb.net.ssl`
- `org.dvb.security.pkcs11`

11.8.6.1.1 The org.dvb.security package

This package provides the AuthProvider class which is needed to log into a PKCS11 token for non key related operations. In that case, there is no other way for a java application to send the PIN code.

It also provides the `KeyStoreBuilder` class to install a callback handler which is called when a PIN code is required to create a `KeyStore`.

This class is a convenience feature to simplify applications and there is also a method to give a password when a `KeyStore` is instantiated. Thus it is possible for an application to log into the token for key operations without using a `KeyStoreBuilder`.

11.8.6.1.2 The org.dvb.auth.callback package

This package is required to support the `KeyStoreBuilder` mechanism. Applications are responsible for asking the end-user for the PIN code. Applications will request the PIN code by implementing the interface `CallbackHandler` from this package. This handler will receive a `PasswordCallback` object when the implementation requires the PIN code.

NOTE: In order to prevent other applications listening in, it is recommended that applications use `org.dvb.event` to obtain exclusive access to the number keys when asking for the PIN code.

11.8.6.1.3 The org.dvb.net.ssl package

During the SSL handshake, the SSL engine uses `TrustManagers` to make sure the certificate chain received from the server is trusted. If client authentication is enabled, the SSL engine will use `KeyManagers` to find a certificate chain and a private key according to the `CertificateRequest` message from the server. The `CertificateRequest` message gives a list of acceptable CA for the server.

In J2SE 5.0 the classes `TrustManagerFactory` and `KeyManagerFactory` can be initialized with a set of `KeyStoreBuilder` where keys and certificate material can be found. To have the same functionality in GEM, the classes `DVBTrustManagerFactory` and `DVBKeyManagerFactory` have been defined. The method init in these classes is used by applications to provide a set of `KeyStoreBuilders` to these factory classes.

Without these two classes, applications can only provide the PIN code at the time the `KeyManagerFactory` is created. Hence if the token is replaced, the existing `KeyManagerFactory` cannot be used and the application would need to create a new instance to use with the new token.

11.8.6.1.4 The org.dvb.security.pkcs11 package

In J2SE 5.0, the Sun PKCS11 provider is configured via a text configuration file. This file is used, to indicate which PKCS11 slot is associated with the provider. For GEM, it may be useful for an application to select a particular slot (if there are several slots and tokens). The methods `getSlotList` and `getTokenInfo` of the class `DVBPKCS11Provider` can be used to get the list of slots and tokens available. The default slot used will be defined by the java property "dvb.security.pkcs11.defaultSlotId". If an application needs to use another slot, it should call the method DVBPKCS11Provider.setSlotId(int slotId) to change the slot used by the provider. The slot can only be changed when the provider is not logged into the token.

11.8.6.2 Specification

This API is formed of the following package and is defined in annex AI, "DVB Extensions for cryptography".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- `org.dvb.security`
- `org.dvb.security.pkcs11`
- `org.dvb.auth.callback`
- `org.dvb.net.ssl`

Installation of cryptographic service providers shall be supported as defined in annex AJ, "Cryptographics service provider installation".

11.9 Other APIs

11.9.1 Timer support

This API is formed of the Timer API defined in Java TV [16] in the `javax.tv.util` package.

Implementations are required to meet the following specifications:

- Minimum repeat interval less than or equal to 40 ms.
- Granularity less than or equal to 10 ms.

The only condition defined in GEM where `TVTimer.scheduleTimerSpec` may throw `TVTimerScheduleFailedException` is if the GEM terminal is unable to provide any more timers. See table 88, "Minimum requirements for other resources".

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

11.9.2 User settings and preferences API

This API is defined in annex L, "User settings and preferences API".

The preferences listed below shall be accessible to all applications.

- User Language.
- Parental Rating.
- Country Code.
- Default Font Size.

Other preferences shall only be accessible where a `UserPreferencePermission` for "read" has been granted (see clause 11.10.2.8, "org.dvb.user.UserPreferencePermission").

11.9.3 Profile and version properties

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

NOTE: This means that a receiver implementing a terminal specification based on the interactive broadcast profile of GEM would return property values consistent with the MHP [1] interactive broadcast profile.

Applications can discover the supported profile and the version of the profile (and thus, what functionality is supported) by retrieving profile and version properties. If a particular profile is not supported then the related version properties shall return null.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

More specifically, the properties listed in table 40 shall be included in the property set of the `java.lang.System` class. Thus these properties can be retrieved using `java.lang.System.getProperty()`. Since this API returns a string, numeric return values shall be encoded as defined by `java.lang.Integer.toString(int)`.

Table 40: System properties for profile and version interrogation

Property	Semantics	Possible values	Example
mhp.profile.enhanced_broadcast	Indicates whether the enhanced broadcast profile is supported	"YES"	"YES"
mhp.profile.interactive_broadcast	Indicates whether the interactive broadcast profile is supported	"YES", null	null
mhp.profile.internet_access	Indicates whether the internet access profile is supported	"YES", null	null
mhp.eb.version.major	Major version number of the supported enhanced broadcast profile	Non-negative integer value	"1"
mhp.eb.version.minor	Minor version number of the supported enhanced broadcast profile	Non-negative integer value	"0"
mhp.eb.version.micro	Micro version number of the supported enhanced broadcast profile	Non-negative integer value	"0"
mhp.ib.version.major	Major version number of the supported interactive broadcast profile	Non-negative integer value	"1"
mhp.ib.version.minor	Minor version number of the supported interactive broadcast profile	Non-negative integer value	"0"
mhp.ib.version.micro	Micro version number of the supported interactive broadcast profile	Non-negative integer value	"0"
mhp.ia.version.major	Major version number of the supported internet access profile	Non-negative integer value	"1"
mhp.ia.version.minor	Minor version number of the supported internet access profile	Non-negative integer value	"0"
mhp.ia.version.micro	Micro version number of the supported internet access profile	Non-negative integer value	"0"
NOTE: In previous versions of the present document, "NO" might be returned instead of null for profiles which are not supported.			

The properties for querying the version of a profile which is not supported by the GEM terminal shall not be supported. Hence, calling `System.getProperty()` for these properties shall return null.

11.9.3.1 Information on options

Clause 15, "Detailed platform profile definitions" defines what is mandatory and optional for each profile. In order to give an application information about which options a particular GEM implementation supports, a property string is defined for each option (with the same granularity as in clause 15).

A GEM implementation supports an option if and only if the corresponding property is known and its value is "SUPPORTED". The properties are part of the property set of the java.lang.System class.

The general syntax of the properties that indicate whether a certain feature is supported or not is:

```
mhp.option.<the optional feature>.
```

The table 41 lists the currently defined options.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 41: System properties for optional feature interrogation

Option	Property
IP Multicast over Broadcast Channel	mhp.option.ip.multicast
HTTP 1.1 over the interaction channel	mhp.option.http.1.1
DSMCC user-to-user over the interaction channel	mhp.option.dsmcc.uu
DVB-HTML	mhp.option.dvb.html
Does this GEM receiver have a smart card reader accessible to applications through the smart card reader API?	mhp.smartcard.reader
Memory for stored services (see clause 9.6.2, "Stored services") is greater than zero (see note)	mhp.stored.services
Memory card access (see clause 11.5.7)	mhp.option.memory.card
OpenType (see clauses 7.4.2 and 11.4.1.5.2)	mhp.option.opentype
High Definition (see clauses 7.2.2, "Video" and G.1.1.3, "High definition")	mhp.option.highdef
Buffered animation (see org.dvb.ui.BufferedAnimation)	mhp.option.buffered.animation
Broadband content guide	mhp.option.broadband.content.guide
Privileged applications	mhp.option.privileged.applications
NOTE: The available memory may be insufficient to store a particular application and hence requests to add applications to stored services may fail even though this property is both known and "SUPPORTED".	

11.9.4 Non-CA smart card API

The API for access to non-conditional smart cards shall be comprised of the following:

- the "SATSA-APDU" optional package defined by SATSA [70];
- the class `javax.microedition.apdu.APDUPermission` defined in clause B.1.2.1 of that document;
- the org.dvb.smartcard package defined in annex AM, "Smart card reader API".

NOTE 1: For GEM terminals without a non-CA smart card reader, the failure mode is defined by SATSA [70].

GEM does not require support for U(SAT).

NOTE 2: SATSA's Generic Connection Framework calls for opening a Connection to a Java Application, identified by an ID (the AID). At the time of this writing, many of the smart cards of interest are standards based but are not Java cards. Although an Application ID could be set on these cards, it is not mandatory and rarely available.

NOTE 3: Any method call of the form Connector.open("apdu...;target=a0.00...") issued against such cards returns a ConnectionNotFound exception. This is because the card has neither an AID set nor is a Java Card with such application listening.

NOTE 4: Any method call of the form Connector.open("apdu...;target=SAT") may also fail as this is normally used on (U)SIM as defined per SIM Application Toolkit mode.

In order to open the APDUConnection with this kind of smartcards, the method openDefaultConnection() in org.dvb.smartcard.SmartCardReader must be used.

In these APDUConnection instances, the Application Selection (SELECT FILE by DF NAME) and MANAGE CHANNEL commands are allowed.

NOTE 5: To open the APDUConnection when the smart card contains AID or it is a (U)SIM, the GCF method can still be used. In this case, Application Selection (SELECT FILE by DF NAME) and MANAGE CHANNEL commands are not allowed.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.9.5 XML parsing API

11.9.5.1 SAX

The XML parsing API is defined in clause 2, "JAXP Subset", of JSR 172 [76].

11.9.5.2 JDOM

This is the org.dvb.xml.jdom package from TS 102 816 [82].

11.9.6 GEM terminal hardware API

This is the org.dvb.hardware package defined in annex AZ, "MHP terminal hardware API".

11.9.7 Content Download API

The content download API is described in Annex BA.

The content download API is based on the shared DVR API for scheduling and managing recording requests [85].

11.10 Java permissions

Under the security model for DVB-J applications, a permissions file is associated with authenticated applications; unauthenticated applications operate within a sandbox environment, which means certain forms of locators will not operate, and certain APIs will not be available. Authenticated applications may be granted permission to use the restricted locators and APIs.

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

This clause explains how the permissions that are defined to be included in the sandbox that is available to unsigned applications and the permissions that can be requested in the permission request file are mapped to the Permission objects in the Java platform.

11.10.1 Permissions for unsigned applications

The GEM security policy includes a set of resources that are always guaranteed to be granted to applications, if the application is executed. Unsigned applications have access to only these resources.

This clause defines the mapping of those resources to the Java Permission objects.

11.10.1.1 java.awt.AWTPermission

This control access to sensitive parts of AWT which is not needed for GEM applications. This shall be denied for both unsigned and signed applications.

11.10.1.2 java.net.SocketPermission

Because access to return channel is not within the sandbox, this is not required for unsigned applications.

11.10.1.3 java.util.PropertyPermission

For unsigned applications, a read permission shall be granted for all properties defined in the present document except for those explicitly identified as only available for signed applications. The permission shall be denied for the action string "write".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A read permission may also be granted for other properties except for those explicitly identified as only available for signed applications.

11.10.1.4 java.lang.RuntimePermission

This permission shall be denied for both unsigned and signed applications.

11.10.1.5 java.io.SerializablePermission

This permission shall be denied for both unsigned and signed applications.

11.10.1.6 java.io.FilePermission

A read permission shall be granted for the subtree under which the implementation mounts the object carousels.

11.10.1.7 javax.tv.media.MediaSelectPermission

The Media API (i.e. JMF) is within the sandbox, so all applications shall be granted a javax.tv.media.MediaSelectPermission with a locator string "*" that indicates access to all media streams.

11.10.1.8 javax.tv.service.ReadPermission

Access to Service Information is within the sandbox, so all applications shall be granted a javax.tv.service.ReadPermission with a locator string "*".

11.10.1.9 javax.tv.service.selection.ServiceContextPermission

All applications shall be granted a `javax.tv.service.selection.ServiceContextPermission` with a name string `"getServiceContentHandlers"` and action string `"own"`.

`ServiceContextPermission( "access", "own" )` shall be granted to all MHP applications.
`ServiceContextPermission( "access", "*" )` shall not be granted.

11.10.1.10 java.util.Locale.setDefault

This method shall throw a security exception.

11.10.1.11 Applications signalled in AIT File

Applications signalled in an AIT File (clause 10.4.5, "AIT File") shall have an instance of `SocketPermission` for each combination of host and port number listed in all transport protocol descriptors with the protocol ID value 0x0003 in both their AIT entry and in the "common" descriptor loop. Each permission shall have the "connect" action only.

11.10.1.12 javax.microedition.xlet.ixc.IxcPermission

Unsigned applications shall be granted:

- IxcPermission("dvb:/unsigned/*", "lookup").
- IxcPermission("dvb:/unsigned/organisation_id/application_id/*", "bind").
- IxcPermission("dvb:/ixc/organisation_id/application_id/*", "bind").
- IxcPermission("dvb:/ixc/*", "lookup").
- IxcPermission("dvb:/service/service_context_id/unsigned/*", "lookup").
- IxcPermission("dvb:/service/service_context_id/unsigned/organisation_id/application_id/*", "bind").

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

organisation_id and application_id refer to the organisation_id and application_id of the application being granted the permission (see clause 10.4.3 "Content of the Application Description"), and shall be formatted as specified in org.dvb.io.ixc.IxcRegistry.lookup() and service_context_id is the platform generated opaque name of the service context (see clause 11.7.3, "Inter-application communication API").

11.10.1.13 MonitorAppPermission and ServiceTypePermission

Unsigned applications shall not be granted MonitorAppPermission or ServiceTypePermission.

11.10.2 Additional Permissions for signed applications

Signed applications can additionally request to get more permissions. These permissions are requested using the permission request file (clause 12.6, "Security policy for applications"). This clause defines the mapping from the items in the permission request file to the Java Permissions that may be granted by the GEM terminal in response to the request.

11.10.2.1 java.util.PropertyPermission

For signed applications, a read permission shall be granted for all properties for unsigned applications and dvb.persistent.root.

11.10.2.2 java.io.FilePermission

A read permission shall be granted for the subtree under which the implementation mounts the object carousels or any other filesystem that may be mounted using the DSMCC APIs.

The following occurs when the permission request file requests the permission to access persistent storage and this is granted:

1) A FilePermission that permits access to the persistent storage directory subtree is created.

NOTE 1: This FilePermission created complies with the access rights defined in clause 12.6.2.7.2, "Policy for signed applications".

2) Where a GEM terminal supports access to memory cards (see clause 11.5.7, "File Storage Device Access"), the implementation shall create instances of java.io.FilePermission allowing access to the locations where the file systems for a memory card will appear in its file system namespace.

When there is a persistent file credential in the permission request file and this is granted, FilePermissions are created as follows:

- file path = value of dvb.persistent.root property + filename from the credential.
- action = string containing "read" if "read" is indicated in the credential; or string containing "write, delete" if "write" is indicated in the credential".

NOTE 2: A single instance of FilePermission is not sufficient to express the limitations on accessing files and directories through credentials required by clause 12.6.2.6, "Credentials". Implementations are responsible for enforcing those requirements and cannot rely on a single instance of FilePermission to achieve this.

11.10.2.3 org.dvb.net.ca.CAPermission

When the permission request file requests the permission to communicate with a CA system and this is granted, a CAPermission is created as follows:

The CA system ID string from the permission request file is directly used as the first part of the string used for the CApermission constructor, this is concatenated with a colon character and the list of the attribute strings based on the attributes listed as true in the permission request file.

This clause does not apply for GEM terminal specifications that do not include the functional equivalent named "Conditional Access" as defined in clause 15.6, "Functional equivalents".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 174 of 861

11.10.2.4 org.dvb.application.AppsControlPermission

When the permission request file requests the permission to have additional permissions to control the lifecycle of applications and this is granted, an `AppsControlPermission` is created.

11.10.2.5 org.dvb.net.rc.RCPermission

When the permission request file requests the permission to communicate through the return channel and this is granted, an `RCPermission` is created as follows:

- For the default ISP item, the `RCPermission` is created with "target:default" string.
- For items with phone numbers in them, the string is "target:" + the phone number prefix in the permission request file + "*".

On `org.dvb.net.rc.ConnectionRCInterface`, the method `getCurrentTarget` shall always throw a `SecurityException`. The method `setTarget` shall throw a security exception where the application doesn't have the permission to use the target specified. The method `setTargetToDefault` shall throw a security exception where the application doesn't have either "target:default" or "target:*" permissions.

11.10.2.6 org.dvb.net.tuning.TunerPermission

When the permission request file requests the permission to access the Tuning API and this is granted, an `TunerPermission` is created.

If a GEM terminal specification does not support the "Tuning API", in clause 11.6.3, listed in table Error! Reference source not found., the class `org.dvb.net.tuning.TunerPermission` is not required by the present document and this clause does not apply.

11.10.2.7 javax.tv.service.selection.SelectPermission

By default a `SelectPermission` is created with a locator "*" and action string "own" and a `ServiceContextPermission` is created with a name "*" and action string "own", unless denied by an entry in the permission request file.

GEM terminal specifications may define types of service other than broadcast. The present document allows replacement of this clause to only apply to broadcast services. Hence, applications will have the permissions needed to select all broadcast services in their own service context(s) unless denied by an entry in the permission request file.

11.10.2.8 org.dvb.user.UserPreferencePermission

When the permission request file requests the permission to read and/or write user preferences and this is granted, a `UserPreferencePermission` is created as follows:

- When the permission request file includes "true" for the "read" attribute and this is granted, a `UserPreferencePermission` is created with the string "read".
- When the permission request file includes "true" for the "write" attribute and this is granted, a `UserPreferencePermission` is created with the string "write".

11.10.2.9 java.net.SocketPermission

When the permission request file requests the permission to communicate with remote hosts and this is granted, `SocketPermissions` are created with the host and action as indicated in the permission request file.

These permissions shall not be granted in MHP terminals where all return channels are represented by instances of `org.dvb.net.rc.ConnectionRCInterface` (i.e. where all return channels are connection oriented) unless an instance of `org.dvb.net.rc.RCPermission` is also granted.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 175 of 861

11.10.2.10 org.dvb.media.DripFeedPermission

When the permission request file requests the permission to use the drip feed feature and this is granted, a `DripFeedPermission` shall be created.

11.10.2.11 org.dvb.application.storage.ApplicationStoragePermission

When the permission request file requests the permission to store applications and this is granted, a ApplicationStoragePermission is created. This permission class is defined in annex AG, "Stored application APIs".

11.10.2.12 javax.microedition.apdu.APDUPermission

When the permission request file requests the permission to access the smart card API and this is granted, an instance of `javax.microedition.apdu.APDUPermission` is created with the name "aid".

11.10.2.13 ServiceContextPermission

On MHP terminals supporting the internet access profile, all applications which are granted a SelectPermission (see clause 11.10.2.7, "javax.tv.service.selection.SelectPermission") shall be granted instances of `javax.tv.service.Selection.ServiceContextPermission` with names:

- "create";
- "destroy";
- "stop";

all of these to have the actions string "own".

11.10.2.14 javax.microedition.xlet.ixc.IxcPermission

Signed applications shall be granted:

- IxcPermission("dvb:/signed/*", "lookup").
- IxcPermission("dvb:/signed/organisation_id/application_id/*", "bind").
- IxcPermission("dvb:/ixc/organisation_id/application_id/*", "bind").
- IxcPermission("dvb:/ixc/*", "lookup").
- IxcPermission("dvb:/service/service_context_id/signed/*", "lookup").
- IxcPermission("dvb:/service/service_context_id/signed/organisation_id/application_id/*", "bind").

`organisation_id` and application_id refer to the `organisation_id` and application_id of the application being granted the permission (see clause 10.4.3 "Content of the Application Description"), and shall be formatted as specified in org.dvb.io.ixc.IxcRegistry.lookup() and service_context_id is the platform generated opaque name of the service context (see clause 11.7.3, "Inter-application communication API").

11.10.2.15 org.dvb.spi.ProviderPermission

Where the permission request file requests the right to install providers and this is granted, an instance of ProviderPermission shall be created for the classname listed. If the scope attribute of the request is "application" then the action of the permission shall be "xlet". If the scope attribute of the request is "system" then the action of the permission shall be "system". Permissions with the action "system" shall only be granted to privileged applications authenticated according to clause 12.18, "Authentication of privileged applications".

11.10.2.16 Permissions for Unbound and Privileged Applications

For signed applications:

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 176 of 861

- When the permission request file requests servicetypepermission and this is granted, an instance of ServiceTypePermission shall be created with the type and action requested in the permission request file.
- When the permission request file requests "monitorapplication" capabilities and this is granted, instances of MonitorAppPermission shall be created for each capability granted.

NOTE 1: See clause 12.18, "Authentication of privileged applications" for requirements on the granting of "monitorapplication" capabilities.

NOTE 2: Clause 10.2.2.2.3 of OCAP [5] (included in the present document by clause 9.13.2, "Service model") defines a number of requirements relating to the granting of Permissions to applications including privileged applications.

11.11 Content referencing

The following mapping shall be used between the types of locator defined in table 59, "Addressable entities, locators and their text representation" and the DVB-J methods defined in this clause. It lists the Java methods and constructors that accept or return (as defined by their method signature) instances of `org.davic.net.Locator`, `javax.tv.locator.Locator`, `javax.media.MediaLocator` or their subclasses. The external form of the locators shall as described in table 59, "Addressable entities, locators and their text representation" for the corresponding entity being referenced. Where the same method is listed as accepting multiple forms of locator, then it is required to accept all forms listed in this clause.

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

11.11.1 Transport stream

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

The term "DVB locators" is considered to refer to all valid locators as described in table 59.

The following methods used in the present document shall accept or return instances of Objects which describe an MPEG transport stream. Methods which accept a locator for a transport stream as an input parameter shall also accept all other DVB locators which include the information to identify a transport stream. If present, service_id, component_tag, event_id and path_segments shall be retained but not used except where there is both a method that accepts such a locator and a query method specified to return that input. In this case the specified semantics of the query method shall be respected.

- `javax.tv.service.transport.TransportStream.getLocator()`.
- `javax.tv.service.transport.TransportStreamCollection.retrieveTransportStream()`.

NOTE 1: This requirement holds only if the terminal supports `TransportStreamCollection` objects.

- `org.davic.net.tuning.StreamTable.getTransportStreams()`.
- `org.davic.net.tuning.NetworkInterfaceController.tune()`.
- `org.davic.net.tuning.NetworkInterface.getLocator()`.
- `org.davic.net.tuning.NetworkInterfaceController.reserveFor()`.
- `org.davic.net.tuning.StreamTable.listTransportStreams()`.
- `org.dvb.si.SITransportStream.getDvbLocator()`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 177 of 861

NOTE 2: For GEM terminal specifications that include the functional equivalent named "Content Referencing" as defined in clause 15.6, "Functional equivalents", the numeric identifiers in the locator are matched in the DVB-SI tables as follows: The numeric identifiers original_network_id and transport_stream_id are matched against the corresponding fields in the NIT. If present, the network_id is matched against the corresponding field in the NIT.

Additionally, `org.davic.net.tuning.NetworkInterface.getLocator()` shall return a DvbNetworkBoundLocator for the network to which the NetworkInterface is connected.

11.11.2 Network

The term "DVB network" is to be interpreted as referring to a valid network, as described in table 59.

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

The following methods used in the present document shall accept or return instances of Objects which describe a DVB network.

- `javax.tv.service.transport.NetworkCollection.retrieveNetwork()`.
- `javax.tv.service.transport.Network.getLocator()`.

11.11.3 Void

11.11.4 Service

11.11.4.1 MPEG/GEM specific service

The following methods used in the present document shall accept or return instances of Objects which describe GEM services and PSI-only services. Methods which accept a locator for a MPEG/GEM service as an input parameter shall also accept all other GEM locators which include the information to identify a service. If present, component_tag, event_id and path_segments shall be retained but not used except where there is both a method that accepts such a locator and a query method specified to return that input. In this case the specified semantics of the query method shall be respected.

- `org.dvb.dsmcc.DSMCCStream.getStreamLocator` where the method `isMPEGProgram` returns true.
- `org.dvb.dsmcc.ServiceDomain.attach()`.
- `org.dvb.dsmcc.ServiceDomain.getLocator()`.
- `org.dvb.application.AppAttributes.getServiceLocator()` where the service is an MPEG / DVB service.
- `org.dvb.dsmcc.ServiceXFRReference` - constructor where the service is a MPEG/DVB service.
- `org.dvb.dsmcc.ServiceXFRReference.getLocator()` - where the service is a MPEG/DVB service.

NOTE: The numeric identifiers in the locator are matched in the DVB-SI tables as follows: the numeric identifiers original_network_id, transport_stream_id (if present) and service_id are matched against the corresponding fields in the SDT. If present, the network_id are matched against the corresponding field in the NIT.

The following methods are not required by the present document:

- `org.davic.net.ca.CAModule.buyEntitlement();`
- `org.davic.net.ca.CAModule.queryEntitlement();`
- `org.dvb.si.SIDatabase.retrieveSIService();`

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

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

The semantics for these methods apply only if the GEM terminal specification requires a given method.

11.11.4.2 Generic service

The following methods used in the present document shall accept or return instances of Objects which reference generic services. These methods are also required to accept or return the same locators as in the previous clause.

- `javax.tv.service.navigation.LocatorFilter` - constructor.
- `javax.tv.service.navigation.LocatorFilter.getFilterValue()`.
- `javax.tv.service.SIManager.getService()`.
- `javax.tv.service.navigation.ServiceDetails.getLocator()`.
- `javax.tv.service.Service.getLocator()`.
- `javax.tv.service.SIManager.retrieveServiceDetails()`.
- `javax.tv.service.navigation.ServiceList.findService()`.
- `org.dvb.application.AppAttributes.getServiceLocator()` - where the service is not a MPEG/GEM specific one.
- `org.dvb.dsmcc.ServiceXFRReference` - constructor where the service is not a MPEG/GEM specific one.
- `org.dvb.dsmcc.ServiceXFRReference.getLocator()` where the service is not a MPEG/GEM specific one.
- `org.dvb.dsmcc.ServiceXFRException` - constructor.
- `org.davic.media.MediaLocator` -constructor.
- `javax.media.MediaLocator` - constructor.
- `org.dvb.locator.NetworkInterfaceBoundMediaLocator` - constructor.
- `javax.media.Manager.createPlayer(MediaLocator)`.
- `javax.media.Manager.createDataSource(MediaLocator)`.
- Constructor of various JMF events consumes JMF `MediaLocators`:
  - `org.dvb.media.CAStopEvent, org.dvb.media.PresentationChangedEvent, org.dvb.media.ServiceRemovedEvent, org.dvb.media.NoComponentSelectedEvent.`
- Various JMF events return JMF `MediaLocators` which were passed into their constructors:
  - `org.dvb.media.CAStopEvent, org.dvb.media.PresentationChangedEvent, org.dvb.media.ServiceRemovedEvent,org.dvb.media.NoComponentSelectedEvent.`
- `StoredApplicationService.getLocator` - return value.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 179 of 861

- `InternetClient.getServiceContentLocators` - return value.

11.11.4.3 Stored services

The following methods used in the present document shall accept or return instances of Objects which describe a stored service.

- `org.dvb.application.storage.StoredApplicationService.getLocator.javax.tv.`

11.11.4.4 Content referencing for IPTV

All the methods listed in clauses 11.11.4.1, "MPEG/GEM specific service" and 11.11.4.2, "Generic service" of the present document shall accept or return instances of Objects which describe unicast and multicast GEM IP services.

- `service.Service.getLocator` - when the Service is a `StoredApplicationService`.

11.11.4.5 Content referencing for OTT services

All the methods listed in clauses 11.11.4.1, "MPEG/GEM specific service" and 11.11.4.2, "Generic service" of the present document shall accept or return instances of Objects, which describe OTT services addressed by HTTP locators.

An HTTPLocator is defined as described in Table 60 and detailed in the Annex AU extension below. This locator references either Media content, a Manifest file as described in clause 7.2.5, ”Streaming Manifest” or an application.

If an HTTP locator references a manifest file, it can be used to obtain a service. This service can then be selected and the platform will perform all necessary steps (manifest parsing, switching from one representation to another based on certain conditions, …).

In GEM terminals where the HTTPS protocol (clause 6.3.7.3, "HTTPS") is supported, an HTTP locator can be used also to address content delivered via HTTPS,

11.11.5 Program event

The following methods used in GEM shall accept or return instances of Objects which describe a program event.

- `javax.tv.service.guide.ProgramEvent.getLocator().`
- `javax.tv.service.SIManager.retrieveProgramEvent().`
- `javax.tv.service.guide.ProgramSchedule.retrieveProgramEvent().`

The following methods are *not* required by GEM:

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

The semantics for these methods only apply if the GEM terminal specification requires a given method.

11.11.6 MPEG elementary stream

The following methods used in the present document shall accept or return instances of Objects which describe an MPEG elementary stream. Methods below which accept as an input parameter an array of locators shall also accept GEM locators including a reference to multiple components.

- `javax.tv.net.InterfaceMap.getLocalAddress().`
- `javax.tv.service.selection.InvalidServiceComponentException` - constructor.
- `javax.tv.media.MediaSelectControl` - all methods accepting or returning instances of javax.tv.locator.Locator.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 180 of 861

- `javax.tv.media.MediaSelectEvent` and subclasses - constructor.
- `javax.tv.media.MediaSelectPermission` - constructor.
- `javax.tv.service.selection.ServiceContext.select()`.
- `org.dvb.dsmcc.ServiceDomain.attach()`.
- `org.dvb.dsmcc.ServiceDomain.getLocator()`.
- `javax.tv.media.MediaSelectControl.getCurrentSelection()`.
- `javax.tv.service.navigation.ServiceComponent.getLocator()`.
- `javax.tv.media.MediaSelectEvent.getSelection()` (also subclasses).
- `org.davic.media.MediaLocator` - constructor - shall also accept multiple component tag GEM locator.
- `org.dvb.locator.NetworkInterfaceBoundMediaLocator` - constructor - shall also accept multiple component tag GEM locator.
- `javax.media.MediaLocator` - constructor - shall also accept multiple component tag GEM locator.
- `javax.media.manager.createPlayer( MediaLocator )` - shall also accept multiple component tag GEM locator.
- `javax.tv.service.selection.InvalidServiceComponentException.getInvalidServiceComponent()`.
- `javax.tv.service.selection.ServiceContentHandler.getServiceContentLocators()`.

NOTE: The numeric identifiers in the locator are matched in the DVB-SI tables as follows: the numeric identifiers original_network_id, transport_stream_id (if present) and service_id are matched against the corresponding fields in the SDT. If present, the network_id are matched against the corresponding field in the NIT.

Methods which accept a locator or an array of locators for an MPEG elementary stream as an input parameter shall also accept all other GEM locators which include the information to identify a MPEG elementary stream. If present, `event_id` and `path_segments` shall be retained but not used except where there is both a method which accepts such a locator as input and a query method specified to return that input. In this case, the specified semantics of the query method shall be respected.

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

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

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

The semantics for these methods only apply if the GEM terminal specification requires a given method.

11.11.7 File

The following methods used in the present document shall accept or return locators which reference files.

- `org.davic.media.MediaLocator` -constructor - for audio files intended to be played from memory.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 181 of 861

- `javax.media.MediaLocator` - constructor - for audio files intended to be played from memory.
- `javax.media.Manager.createPlayer(MediaLocator)` - for audio files intended to be played from memory.
- `javax.media.Manager.createDataSource(MediaLocator)` - for audio files intended to be played from memory.
- `org.dvb.dsmcc.ServiceDomain.getURL(Locator)` - locators referring to File or Directory entities, as defined in table 59.

Apart from the above, all file references shall be encapsulated in instances of java.net.URL.

11.11.8 Directory

The following method shall return an instance of the GEM locator referencing a directory.

- `org.dvb.application.AppIcon.getLocator()`

11.11.9 Drip feed decoder

The following methods used in the present document shall accept or return locators which reference the "drip feed" decoder.

- `javax.media.MediaLocator` - constructor
- `javax.media.Manager.createDataSource(MediaLocator)`

11.11.10 Void

11.11.11 Methods working on many locator types

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

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

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 182 of 861

11.11.12 Support for the HTTP Protocol in DVB-J

In GEM terminals where an HTTP protocol (clause 6.3.7.1, "HTTP 1.1" or clause 6.3.7.2, "GEM profile of HTTP 1.0") is supported, the following classes and methods shall support the HTTP protocol concerned. In GEM terminals where the HTTPS protocol (clause 6.3.7.3, "HTTPS") is supported, the following class and methods shall support that protocol.

- The constructor for javax.media.MediaLocator - for referencing audio files intended to be played from memory.
- Methods on javax.media.Manager accepting javax.media.MediaLocator as input parameters - for constructing JMF players for audio files intended to be played from memory.
- Methods in the present document which accept instances of java.net.URL are required to accept instances which encapsulate an "http" URL and behave according to their specification. - e.g org.havi.ui.HSound.load(java.net.URL).
- On GEM terminals supporting applications downloaded over the interaction channel as defined in clause 9.6 "Services and applications not related to conventional DVB services", the method `LocatorFactory.createLocator(String)` shall additionally accept Strings containing URLs using the "http" and "https" protocols as being valid and return a corresponding Locator. This method shall only validate the string to the extent that this is possible without network access.
- On GEM terminals supporting applications downloaded over the interaction channel as defined in clause 9.6 "Services and applications not related to conventional DVB services", the method `SIManager.getService(Locator)` shall accept such Locators as being valid and return a corresponding `javax.tv.service.Service`. This method shall only validate the locator to the extent that this is possible without network access.

When HTTP URLs are used with instances of DVBClassLoader to load DVB-J classes over the interaction channel in a signed application, the requirements of the GEM security model shall be complied with before a class is allowed to be successfully loaded from such a URL.

11.11.13 GEM Applications

The following methods used in the present document shall accept or return instances of Locators which identify a GEM application:

- `javax.tv.service.selection.ServiceContext.select(Locator [] ).`

11.12 Stand-alone Applications

11.12.1 Common behavior

The extended GEM application model allows several forms of applications to execute in a stand-alone mode independent of any broadcast services. The environment of these stand-alone applications shall be as follows:

- Instances of `org.dvb.si.SIDatabase` may not return any SI objects if the corresponding network interface isn't tuned to anything. It is implementation dependent whether some network interfaces are left tuned to some previously used transport stream.

For DVB-J applications, the behaviour defined in clause 9.6.4, "Common behaviour" shall mean the following:

- The method `ServiceContext.getServiceContentHandlers` shall not return a `ServiceContentHandler` for any such existing video or audio for a stand-alone stored application.
- A stand-alone application creating and starting a JMF player shall receive a higher priority for resources than the presentation of any previously existing video and audio.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.12.2 Stored services

The GEM specification supports two forms of stored applications as defined in clause 9.1.9, "Cached applications" and clause 9.6.2, "Stored services".

11.12.2.1 Stored application APIs

This is defined in annex AG, "Stored application APIs" of the present document.

11.12.2.2 Modified behaviour of GEM 1.0 APIs

The following APIs defined in GEM 1.0 shall have extended semantics to support listing and launching of stored services as defined in clause 9.6.2, "Stored services". The modifications to these APIs are as follows:

- The class javax.tv.service.navigation.ServiceTypeFilter when used with a ServiceType "`org.dvb.application.StoredApplicationServiceType.STORED_APPLICATION_SERVICE`" shall return a `ServiceList` containing all currently installed stored services when passed to SIManager.`filterServices`.

NOTE: Implementations may choose to expose other installed services (e.g. manufacturer resident ones) through this API.

- Instances of `javax.tv.service.Service` returned from such a `ServiceList` shall be instances of `org.dvb.application.StoredApplicationService` with the modified semantics as defined for that interface.
- The method `ServiceContext.select(Service selection)` shall accept instances of `org.dvb.application.StoredApplicationService` and attempt to start the stored application service concerned subject to the GEM security model.
- Attempting to select a stored application service which contains no autostart applications shall fail with a `SelectionFailedEvent` with reason code `CONTENT_NOT_FOUND`.
- The file system APIs shall see a file system comprised of the files defined in the application description file as needing to be stored for this application. Calling new `DSMCCObject(".")` or new `java.io. File(".")` shall instantiate the directory object that refers to the outermost (top level) directory listed in the application description file (see clause 10.8.3, "Application description file") when the application was stored. Other file system behaviour shall be as defined in clause 9.6.2, "Stored services".
- The `org.dvb.dsmcc.DSMCCObject` class shall work for access to files within a stored application. The methods with modified semantics in these circumstances are as follows:
  - `abort()` - shall always throw a `NothingToAbortException`.
  - `asynchronousLoad(AsynchronousLoadingEventListener)` - if the file exists, succeeds immediately (with `SuccessEvent` being generated) otherwise fails with `InvalidPathNameException`.
  - `isObjectKindKnown()` - always returns true.
  - `isStream()` - always returns false.
  - `isStreamEvent()` - always returns false.
  - `loadDirectoryEntry(AsynchronousLoadingEventListener)` - always succeeds immediately with `SuccessEvent` being generated.
  - `prefetch(both signatures)` - always returns false.
  - `setRetrievalMode(int)` - silently ignored.
  - `synchronousLoad()` - if the file exists, succeeds immediately, otherwise fails with `InvalidPathnameException`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Additionally:

- `ObjectChangeEvents` shall not be generated.
- The `unload()` method shall not be considered as removing stored files from where they are stored.
- Access to files shall not fail for reasons of a service domain not being in an attached state, even though a stored filesystem cannot be represented by a DSMCC Service Domain.
- `File.getCanonicalPath()` and `File.getAbsolutePath()` cannot be relied upon to be return the same strings for the broadcast and stored cases.
- Within a single application instance, the absolute paths returned by `java.io.File.getAbsolutePath()` shall be accepted by constructors for `java.io.File` and sub-classes to reference the same underlying file as the one on which `getAbsolutePath()` was called.

11.12.2.3 Permissions

11.12.2.3.1 FilePermission

A read permission shall be granted for the subtree corresponding to the outermost (top level) directory listed in the application description file and recursively all directories below that.

11.12.2.4 Stored application management API

This is the org.dvb.application.privileged package defined in annex AT, "Application Management API".

11.13 Void

11.14 Internet Access

11.14.1 Internet client control APIs

These are defined in the `org.dvb.internet` package (see annex AH, "Internet client APIs").

Internet clients accessible to GEM applications shall appear in the list of services maintained by the `SIManager` which is returned by calls to the method `SIManager.filterServices`. For each accessible internet client, the corresponding sub-class of `InternetClientService` shall be returned from calls to that method both when passed an instance of `ServiceTypeFilter` constructed with the type `InternetServiceType.INTERNET_CLIENT` and when passed null.

The method `ServiceContext.select(Service selection)` shall accept instances of `InternetClientService` and its sub-classes returned by the mechanism described above as being valid input.

NOTE 1: When a internet client application is selected in the same service context as a GEM application, the mechanism by which that internet client application terminates is implementation dependent. Examples of termination may include the end-user closing the browser (by some mechanism), the end user selecting a broadcast TV service using the navigator or the end user selecting a broadcast TV service by selecting a link to a "dvb:" URL in a WWW page or email. See also clause 9.7, "Lifecycle of internet access applications".

GEM terminals capable of supporting GEM applications and internet client applications simultaneously shall support the creation of a second service context by GEM applications using `ServiceContextFactory.createServiceContext()` in addition to the one used for "normal" GEM applications. This second service context need only support the selection of instances of `InternetClientService`. Selection of other services in this second service context would fail with a `SelectionFailedEvent` with reason code `INSUFFICIENT_RESOURCES`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 185 of 861

NOTE 2: This does not prevent GEM terminals with multiple independent video outputs supporting a service context per video output capable of supporting GEM services however in this case, these service contexts would be created by the GEM navigator.

NOTE 3: Support of multiple service contexts on the same video output each allowing presentation of GEM services is not excluded in the present document but is not fully specified here.

11.14.2 Internet applet support

Internet applet support is defined as follows.

11.14.2.1 HTML tags

Internet applets shall be supported using the "<applet>" and "<object>" tag with the semantics defined in HTML 4 [69].

11.14.2.2 Java Platform

The Java platform for Internet applets shall be PP 1.1 [77].

11.14.2.3 Void

11.14.2.4 Void

11.14.2.5 Security

The present document does not require support for signed applets or for the code signing mechanism and APIs as defined in PP 1.1 [77].

If applets can access Java APIs defined as part of clause 11, "DVB-J platform", apart from those listed in clause 11.14.2, "Internet applet support", they shall have the permissions required to be available to unsigned GEM applications. Of the permissions available to signed GEM applications, they shall have only the following:

- `java.util.PropertyPermission` to `"read"` all properties defined for the `java.lang.system.getProperties()` method.
- `java.net.SocketPermission` to `"connect"` to all ports on the server from which the applet was originally downloaded.

11.15 APIs defined in OCAP

This clause is optional in all profiles of GEM. GEM terminal specifications may include all or part of the specification elements of this clause, or may include none at all. If specification elements are included, functional equivalents may be specified where necessary.

11.15.1 Introduction (informative)

The present document includes a number of APIs defined in OCAP [5] but not all of them. Many classes in OCAP are wholly or substantially specific to the US cable TV environment and would not be usable elsewhere. The following table summarize the classes included from OCAP and their relation with the present document.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 42: Classes from OCAP in the present document

Name of class	Included in present document	Not included in present document	
		Specific to US cable environment	Other reasons
org.ocap.application.AppFilterHandler	Yes		
org.ocap.application.AppFilter	Yes		
org.ocap.application.AppManagerProxy	Partially		
org.ocap.application.AppPattern	Yes		
org.ocap.application.AppSignalHandler	Yes		
org.ocap.application.OcapAppAttributes	Partially		
org.ocap.application.PermissionInformation	Yes		
org.ocap.application.SecurityPolicyHandler	Yes		
org.ocap.event.EventManager			Yes
org.ocap.event.UserEventAction			Yes
org.ocap.event.UserEventFilter			Yes
org.ocap.event.UserEvent			Yes
org.ocap.hardware.CopyControl		Yes	
org.ocap.hardware.Host		Yes	
org.ocap.hardware.IEEE1394Node		Yes	
org.ocap.hardware.pod.HostParamHandler		Yes	
org.ocap.hardware.pod.PODApplication		Yes	
org.ocap.hardware.pod.POD		Yes	
org.ocap.hardware.PowerModeChangeListener			Yes
org.ocap.hardware.VideoOutputPort		Yes	
org.ocap.media.AlternativeMediaPresentationEvent			Yes
org.ocap.media.AlternativeMediaPresentationReason			Yes
org.ocap.media.ClosedCaptioningAttribute		Yes	
org.ocap.media.ClosedCaptioningControl		Yes	
org.ocap.media.ClosedCaptioningEvent		Yes	
org.ocap.media.ClosedCaptioningListener		Yes	
org.ocap.media.FilterResourceAvailableEvent		Yes	
org.ocap.media.ForcedDisconnectedEvent		Yes	
org.ocap.media.MediaAccessAuthorization			Yes
org.ocap.media.MediaAccessConditionControl			Yes
org.ocap.media.MediaAccessHandler			Yes
org.ocap.media.MediaAccessHandlerRegistrar			Yes
org.ocap.media.MediaPresentationEvaluationTrigger			Yes
org.ocap.media.MediaPresentationEvent			Yes
org.ocap.media.MediaTimer			Yes
org.ocap.media.MediaTimerListener			Yes
org.ocap.media.NormalMediaPresentationEvent			Yes
org.ocap.media.NotPresentedMediaInterface			Yes
org.ocap.media.VBIFilterEvent		Yes	
org.ocap.media.VBIFilterGroup		Yes	
org.ocap.media.VBIFilter		Yes	
org.ocap.media.VBIFilterListener		Yes	
org.ocap.mpeg.PODExtendedChannel		Yes	
org.ocap.net.OcapLocator		Yes	
org.ocap.net.OCRCInterface		Yes	
org.ocap.OcapSystem	Yes		
org.ocap.resource.ApplicationResourceUsage			Yes
org.ocap.resource.ResourceContentionHandler			Yes
org.ocap.resource.ResourceContentionManager			Yes
org.ocap.resource.ResourceUsage			Yes
org.ocap.service.AbstractService	Yes		
org.ocap.service.AbstractServiceType	Yes		
org.ocap.service.ServiceContextResourceUsage			Yes
org.ocap.service.ServiceTypePermission	Yes		
org.ocap.si.Descriptor			Yes
org.ocap.si.DescriptorTag		Yes	
org.ocap.si.PATProgram		Yes	
org.ocap.si.PMTElementaryStreamInfo		Yes	
org.ocap.si.ProgramAssociationTable		Yes	
org.ocap.si.ProgramAssociationTableManager		Yes	

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 187 of 861

Name of class	Included in present document	Not included in present document	
		Specific to US cable environment	Other reasons
org.ocap.si.ProgramMapTable		Yes	
org.ocap.si.ProgramMapTableManager		Yes	
org.ocap.si.StreamType		Yes	
org.ocap.si.TableChangeListener			Yes
org.ocap.si.Table			Yes
org.ocap.storage.DetachableStorageOption	Yes		
org.ocap.storage.ExtendedFileAccessPermissions	Yes		
org.ocap.storage.LogicalStorageVolume	Yes		
org.ocap.storage.RemovableStorageOption	Yes		
org.ocap.storage.StorageManagerEvent	Yes		
org.ocap.storage.StorageManager	Yes		
org.ocap.storage.StorageManagerListener	Yes		
org.ocap.storage.StorageOption	Yes		
org.ocap.storage.StorageProxy	Yes		
org.ocap.system.EASHandler		Yes	
org.ocap.system.EASModuleRegistrar		Yes	
org.ocap.system.event.DeferredDownloadEvent	Yes		
org.ocap.system.event.ErrorEvent	Yes		
org.ocap.system.event.RebootEvent	Yes		
org.ocap.system.event.ResourceDepletionEvent	Yes		
org.ocap.system.event.SystemEvent	Yes		
org.ocap.system.event.SystemEventListener	Yes		
org.ocap.system.event.SystemEventManager	Yes		
org.ocap.system.MonitorAppPermission	Partially		
org.ocap.system.RegisteredApiManager			Yes
org.ocap.system.SystemModuleHandler		Yes	
org.ocap.system.SystemModule		Yes	
org.ocap.system.SystemModuleRegistrar		Yes	
org.ocap.test.OCAPTest		Yes	
org.ocap.ui.event.OCRcEvent			Yes
org.ocap.ui.HSceneBinding			Yes
org.ocap.ui.HSceneChangeRequestHandler			Yes
org.ocap.ui.HSceneManager			Yes

11.15.2 OCAP Annex G - the org.ocap.application package

This package shall be supported with the following modifications.

- For OcapAppAttributes, the following language shall not apply and the behaviour specified for the methods in GEM shall be supported.

  For this version of the OCAP profile, the getProfiles() method always returns ocap.profile.basic_profile

- For OcapAppAttributes, the following language shall only apply to applications signalled in an XAIT or added to the AppsDatabase by registerUnboundApp. It shall not apply to applications signalled in an AIT.

  org.dvb.application.AppsDatabase must return an instance of OcapAppAttributes by the getAppAttributes methods.

- The method AppManagerProxy.setApplicationPriority shall have no effect.

- The method AppManagerProxy.registerUnboundApp shall accept as valid input XAITs formatted in both the MPEG-2 section and table format as specified in OCAP and the XML format as specified in clause 10.16.2, "XAIT" of MHP [1].

11.15.3 OCAP Annex P - the org.ocap.service package

This package shall be supported except for ServiceContextResourceUsage.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

11.15.4 OCAP Annex Q - the org.ocap.system package

From this package, MonitorAppPermission shall be supported. The following permission names shall be supported.

- "registrar".
- "servicemanager".
- "service".
- "handler.appFilter".
- "security".
- "reboot".
- "systemevent".

11.15.5 OCAP annex O - the org.ocap package

This package shall be supported. The requirements relating to initialising the APIs for conformance testing do not apply since those APIs are not required by the present document.

11.15.6 OCAP annex U - the org.ocap system.event

The org.ocap.system.event package shall be supported.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12 Security

12.1 Introduction

This clause covers the following areas of security:

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

12.1.1 Overview of the security framework for applications

The security framework enables a receiver to authenticate the source of application code or other files. In the case of application code files, the authentication advises the receiver what access rights should be granted to an application for sensitive resources, see clause 12.6, "Security policy for applications" for more detail.

The system uses 3 different authentication messages:

- Cryptographic hash codes:
  - This provides a summary of a quantity of data - typically a subset of the total set of data under consideration.
- Signatures:
  - These deliver a master hash code (computed over all of the appropriate data) that has been "signed" by an authorising organization. The signing process securely associates the master hash code with the signatory. The hash code process shows that the data has not been tampered with since it was signed by the signatory.
- Certificates:
  - These provide a "chain of trust" from the authorising organization up to some trusted third party (the root certificate authority) that is well known to the receiver.

The messages are delivered within files of the file system so this authentication scheme is applicable to any hierarchical file system whether operating over the broadcast or return channels.

Applications that are eligible to be trusted shall be identified with an application_id from range of values allocated for signed applications (see clause 10, table 13, "Value ranges for application_id"). Applications that are not eligible to be trusted shall be identified with an application_id from the unsigned applications range. For an application with an application_id from the signed application range that requires signing to establish trust, if it is not signed it is considered to have failed authentication. An application with an application_id from the unsigned application range is treated as not eligible to be trusted even if the files might be transmitted with signatures.

12.1.2 Overview of return channel security

In this version of the specification general purpose protocols and a standard cryptographic suite derived from internet standards are used.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12.1.3 Establishing trusted applications

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

Any security framework, whether it involves codesigning or not, shall:

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

12.2 Authentication of applications

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

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

12.2.1 Overview of authentication messages

Three different message types are used: "Hash codes", "Signatures" and "Certificates". Each message is placed in a file. The placement of the files depends on their function and is specified under the appropriate headings under clause 12.4, "Detail of application authentication messages".

12.2.1.1 Hash codes

The present document describes application of hash codes to the following types of information:

- Files.
- Directories.

The hash computation considers the content and attributes of the objects rather than transport specific information. As a result, the authentication is independent of the underlying transport protocol.

In the case of a directory the hash value depends on the hash values of the objects bound to it, and so provides a hash of all of the objects to be authenticated in the "tree" below it.

12.2.1.2 Signatures

The data authenticated is a hierarchical file system (for example DSM-CC OC). The root of an authenticated "tree" carries one or more signatures. This allows one or more organizations to sign a set of information.

The root of the authenticated "tree" can be the root directory of the file system or the "top" directory of a "subtree".

The signature:

- References a certificate containing the public key required to decode the signature.
- Identifies the hash algorithm used.
- And the value of the signature.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 191 of 861

12.2.1.3 Certificates

The certificate provides a public key that can be used to decode a hash code contained in a signature and so enable a tree/subtree to be verified. The certificate itself is signed by a higher certification authority.

To correctly authenticate a subtree there must be a valid "chain" of certificates from the signature to a root certificate as is illustrated in figure 14.

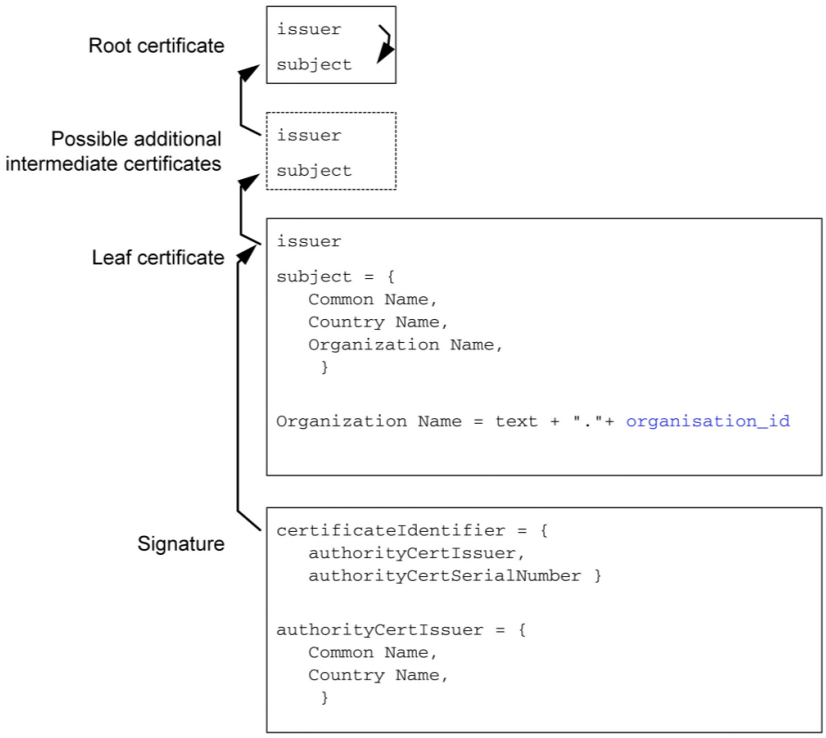


Figure 14: Certificate chain illustrated

12.2.1.4 Authentication of hierarchical file systems

The solution here is based on authentication of a hierarchical structure of objects. Hashcodes are computed systematically and accumulatively across some or all of the objects in the hierarchy. A signature at the top of the hierarchy identifies the source of the objects.

The framework provides a flexible and non-time consuming method enabling the authentication of subtrees of a file system with a single signature. Since checking signature is far more time-consuming than checking hashcode values, this mechanism is more efficient than signing each object of a subtree.

Further, only the objects that are loaded need real-time hashcode checking.

This mechanism does NOT mandate that the whole subtree is authenticated.

NOTE: The broadcaster can choose which files in the file system are authenticated. For example code files might be authenticated and asset files might be left without authentication.

Finally, this framework embraces key distribution by specifying a certificate mechanism, the aim of this is to certify that the key used to compute the signature is valid and used by a certified service provider.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12.3 Message transport

The authentication messages are transported in files.

In no cases shall a service transfer be required to access the file content. In the case that the file system is an object carousel this means that the IOR for the authentication message files shall always use a BIOP profile body and never a Lite options profile body.

12.4 Detail of application authentication messages

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

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

Three data structures are defined for communicating authentication information:

- HashFile.
- SignatureFile.
- CertificateFile.

These are placed in files in the file system. The location of the file depends on its function.

12.4.1 HashFile

12.4.1.1 Description

The HashFile lists all of the elements of the current directory except itself, and possibly except signature files as described in the following. On transport protocols supporting the listing of files in a directory (e.g. Object Carousel), signature files shall either all be listed with a digest type of non-authenticated or shall all be omitted. On transport protocols not supporting the listing of files in a directory (e.g. HTTP), signature files shall be listed and shall have a digest_type of non-authenticated.Those elements to be authenticated are associated with hashcodes. The syntax of the HashFile is shown in table 43.

Table 43: Syntax of the Hashfile

Syntax	Num. Bits	Format
Hashfile () {		
digest_count	16	uimsbf
for(i=0; i<digest_count; i++) {		
digest_type	8	uimsbf
name_count	16	uimsbf
for(j=0; j<name_count; j++) {		
name_length	8	uimsbf
for(k=0; k<name_length; k++) {		
name_byte	8	bslbf
}		
}		
for(j=0; j<digest_length; j++) {		
digest_byte	8	bslbf
}		
}		
}		
Other data may follow but can be ignored by implementations conforming to this profile.		

digest_count: This 16 bit value identifies the number of digest values in this hash file.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

digest_type: This 8 bit value identifies the digest computation rules and the digest algorithm, if any, used for the associated objects. Table 44 lists the allowed values for this field. The digest computation rules are defined in clause 12.4.1.3, "Digest value computation rules".

Table 44: Values of digest_type

Value	Digest len.	Aalgorithm
0	0	Non authenticated
1	16	Digest computation rules without prefix and with MD-5 as defined in RFC 1321 [44]
2	20	Digest computation rules without prefix and with SHA-1 as defined in FIPS-180-1 [57]
3	20	Digest computation rules with prefix and with SHA-1 as defined in FIPS-180-1 [57]
Other values		Reserved for future use

name_count: This 16 bit value identifies the number of object names associated with the digest value. The value of this field shall be greater than zero.

name_length: This 8 bit value identifies the number of bytes in the object name.

name_byte: This 8 bit value holds one byte of the object name.

Each name shall be the name of an object in the directory that contains the HashFile. So, file names are the names of files in the directory and directory names are the names of direct sub-directories of the directory. No path information shall be included in the name.

The names carried by this field are binary identical to the payload part of names in the file system. So, any name matching process can be binary and ignorant of character encoding, letter case etc. Also, terminating null characters are not considered to be part of the file name.

digest_length: This integer value gives the number of bytes in each digest value. It depends upon the digest type as tabulated in table 44. GEM terminals shall support all digest algorithms.

NOTE: Non-authenticated objects have a zero length digest.

digest_byte: This 8 bit value holds one byte of the digest value. See clause 12.4.1.3, "Digest value computation rules".

12.4.1.2 HashFile location and naming conventions

An application comprises files containing data that can be spread across various directories and is contained within a subtree of the file hierarchy. A HashFile will be put in each directory containing objects that need to be authenticated.

The name of the HashFile shall be:

```
"dvb.hashfile"
```

There shall only be one instance of HashFile per directory that contains authenticated resources.

See clause 12.7, "Example of creating an application that can be authenticated".

12.4.1.3 Digest value computation rules

The digest value is computed over the objects named in the HashFile in the order listed in the HashFile. The length of the list of objects associated with each digest value may be one or more.

Each list of objects may contain an arbitrary mix of different object types (e.g. a mixture of file and directory names). The digest value is computed by first initialising the digest algorithm in an algorithm specific way and then applying the relevant data for each object to the algorithm in order. The relevant data for each object depends on its type (File or Directory) and on the value of digest_type and is specified in table 45.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 194 of 861

Table 45: Data required for digest value computation

Object type	Digest type	entry_type	Relevant data
File	1 or 2	not applicable	The entire content of the file.
Directory			The content of the HashFile of the named directory.
File	3	1	A prefix concatenated with the entire content of the file. The prefix is made of the entry_type encoded as a 32 bit uimsbf and concatenated with the file length in bytes encoded as a 32 bit uimsbf.
Directory		0	A prefix concatenated with the content of the HashFile of the named directory. The prefix is made of the entry_type encoded as a 32 bit uimsbf and concatenated with the file length in bytes encoded as a 32 bit uimsbf.
NOTE: All other values of entry type are reserved for future use.			

12.4.1.3.1 Example

Consider two files `file1` and `file2` and one directory `dir1`. Using digest type 3, the digest value of `file1` + `file2` + `dir1` is equal to:

```
SHA-1 ( (uimsbf 32) 1 + (uimsbf 32) FileLength ( file1) + contents of file1
+ (uimsbf 32) 1 + (uimsbf 32) FileLength(file2) + contents of file2
+(uimsbf 32) 0 +(uimsbf 32) FileLength(HashFile(dir1))
+ contents of HashFile(dir1) ).
```

12.4.1.4 Warning concerning grouping of objects under a single digest (informative)

Broadcasters should be made aware that during the construction of the object carousel, grouping objects under one digest can significantly and adversely affect the performance and memory requirements of an application on a terminal. For example, if three objects are grouped under the same digest, implementations must load all three objects even if only one is needed.

GEM in general recommends that while setting up a signed application for broadcast, only one digest be associated with each object requiring authentication. Grouping of multiple files under one hash value should only be employed if those files must always be loaded together and simultaneously. Additionally, the rate of change of the contents of an object should be carefully considered because grouping a rapidly changing object with more slowly changing objects for the purposes of hashing will negatively impact performance. When attempting such grouping, the resource limits of the GEM terminal should always be carefully considered.

12.4.1.5 Special authentication rules

a) For objects which are directories, if the digest_type is non-zero there shall be a HashFile in the sub-directory listed. If the HashFile is absent then the authentication fails.

b) Each HashFile shall provide a complete list of all the objects named in the directory except itself and any possible signature files mentioning each name exactly one time. The behaviour varies depending on the ability of the transport protocol to support directory listing.

- On transport protocols supporting the listing of files in a directory (e.g. Object Carousel), the authentication shall fail if the set of objects listed in the HashFile is different to the set of objects in the directory.

- On transport protocols that don't support the listing of files in a directory (e.g. HTTP), the set of names in the HashFile acts as a filter for the set of objects that can be accessed. Attempts to access a file neither named in the HashFile nor the HashFile itself, nor the signature files shall fail (that is behave as if the file is not available). The file system for loading applications from the interaction channel via HTTP is a case of such a file system, see clause 6.4.1.4, "Directory listing in this file system".

These rules apply regardless of the value of digest_type associated with the object.

c) If the digest_count is equal to 0, the file HashFile shall be ignored. Every entry in the directory will be non authenticated.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

d) If a name_length is equal to zero (or if a name is not found in the directory containing the HashFile), all the names associated with the current digest value shall be considered as incorrectly authenticated. It means the authentication checking will fail for these entries.

e) If the object is a Stream or StreamEvent then the digest_type shall be zero.

f) Objects associated with a digest_type that the receiver does not support shall be treated by that receiver as if the value of digest_type was zero (not authenticated).

g) There is no requirement for certificates files to have a digest_type other than zero (not authenticated). However, if they do have a non-zero digest_type they don't receive exceptional treatment.

NOTE: There is no security benefit in including a certificate file with a digest_type other than zero.

h) When a permission request file exists, its entry in the HashFile cannot have a digest type of 0. GEM terminals shall ignore a permission request file with a digest type of 0.

12.4.2 SignatureFile

12.4.2.1 Description

The SignatureFile is a File containing one digital signature. It contains the following ASN.1 DER structure:

```
Signature  ::=  SEQUENCE  {
    certificateIdentifier                    AuthorityKeyIdentifier,
    hashSignatureAlgorithm                   OBJECT IDENTIFIER,
    signatureValue                           BIT STRING  }
```

certificateIdentifier : As defined in the ITU-T Recommendation X.509 [52] extension for the AuthorityKeyIdentifier field. It identifies the certificate that carries the certified public key that is used to check the signature.

```
AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier                            [0] KeyIdentifier OPTIONAL,
    authorityCertIssuer                    [1] GeneralNames OPTIONAL,
    authorityCertSerialNumber                [2] CertificateSerialNumber OPTIONAL }
```

Implementations are not required to use the possibly present `keyIdentifier` element of the `AuthorityKeyIdentifier`. The `AuthorityKeyIdentifier` structure shall contain both the `authorityCertIssuer` and `authorityCertSerialNumber` elements.

The `authorityCertIssuer` shall contain the field "`directoryName`", this field shall be equal to the `issuerName` of the certificate that carries the public key used to check the signature.

hashSignatureAlgorithm: this field identifies the hash algorithm that is used.

NOTE: The encryption algorithm used to compute the signature is already described in the SubjectKeyInfo field of the certificate that certifies this key, and thus that only the identification of the hash algorithm is needed. The supported algorithms are MD5 and SHA-1.

```
md5 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) US(840) rsadsi(113549)
    digestAlgorithm(2) 5 }

sha-1 OBJECT IDENTIFIER ::=
    { iso(1) identified-organization(3) oiw(14) secsig(3)
    algorithm(2) 26 }
```

signatureValue: See clause 12.11.1.3, "signatureValue".

12.4.2.2 SignatureFile location and naming conventions

The SignatureFile is located in the root directory of the subtree that it signs. There can be several SignatureFiles, as there can be several entities that sign the structure. See clause 12.4.4, "Integration".

By convention, the name of a SignatureFile is:

```
"dvb.signaturefile."<x>
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 196 of 861

where the <x> is a textual representation of an integer decimal number without leading zeroes. The range of values represented in any single directory shall start with 1 and increment in steps of 1. The first unused integer value in the ascending sequence indicates the end of the range.

The purpose of this x is to allow the hash file of an authenticated subtree to be signed by more than one entity. It is equal to the x value in the file name of the corresponding certificate file. See clauses 12.4.3.5, "CertificateFile location and naming conventions" and 12.4.4, "Integration".

12.4.2.3 Supported algorithms

Signing data is a two-step process:

a) First a hash is computed over the data.

b) The resulting hashvalue is then encrypted using an encryption algorithm.

As indicated in clause 12.4.1, "HashFile," GEM defines two possible hash algorithms: MD5 and SHA-1.

The encryption algorithm used to compute the signature is indicated in the certificate that carries this key.

12.4.2.4 Signature computation rules.

The hash is computed over the content of the HashFile contained in this root directory.

NOTE: This is the same principle as for the classical hash computation described in clause 12.4.1.3, "Digest value computation rules".

12.4.2.5 Authentication rules

See signatureValue.

12.4.3 CertificateFile

12.4.3.1 Description

The CertificateFile contains all of the certificates in the certificate chain up to, and including, the root certificate. The leaf certificate is placed first in the file. The last certificate in the file is the root certificate. The root certificate is included in this file only for consistency. How the GEM terminal determines its policy related to this root CA is implementation dependent. The encoding of the certificate is defined in ITU-T Recommendation X.509 [52]. Below is defined the profile of ITU-T Recommendation X.509 [52] for use in authenticating GEM applications. This profile is based on RFC 2459 [55].

The syntax of the CertificateFile is shown in table 46.

Table 46: Syntax of the CertificateFile

Syntax	Num. Bits	Format
Certificatefile () {		
certificate_count	16	uimsbf
for(i=0; i<certificate_count; i++) {		
certificate_length	24	uimsbf
certificate()		
}		
}		

certificate_count: This 16-bit integer carries the number of certificates in the certificate file.

certificate_length: This 24-bit integer specifies the number of bytes in the certificate.

certificate(): This field carries a single g data structure as defined by ITU-T Recommendation X.509 [52]. See clause 12.11.1, "Main part of the certificate".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 197 of 861

12.4.3.2 ASN.1 encoding

The basic specification of the ASN.1 DER encoding used in RFC 2459 [55] is given in ASN.1 [54]. However, RFC 2459 [55] defines some extensions which are required to implement GEM.

12.4.3.3 Supported algorithms

There are various algorithm identifiers in the certificate structure. The OID of the algorithm used in the SubjectPublicKeyInfo structure shall be RSA.

The values for AlgorithmIdentifier used both in the certificate structure and in the TBSCertificate structure that are supported in GEM are listed under clause 12.5.1, "signatureAlgorithm".

12.4.3.4 Name matching

The only allowed encoding of attributes of distinguished names shall be UTF8String.

NOTE: The use of this encoding allows name matching to be a binary comparison.

12.4.3.5 CertificateFile location and naming conventions

As described in clause 12.2.1.3, "Certificates", a key can be authenticated through a "certificate chain".

A certificate chain is a hierarchy of certificates that enable the implementation to verify the validity of the key used to check a signature. However, for consistency, the file shall carry all of the certificate chain up to, and including, the root certificate.

The certificate file that leads to the public key of a signature shall be placed in the same directory as that signature file.

The name of a CertificateFile is:

```
'dvb.certificates.'<x>
```

where the <x> corresponds to the x value in the file name of the signature file verified by this certificate. Hence in the root directory of an authenticated subtree there shall be one certificate file for each signature file. See clauses 12.4.2.2, "SignatureFile location and naming conventions" and 12.4.4, "Integration".

12.4.3.6 Authentication rules

Certificates are considered to be the same if they have bitwise identical contents.

Where a GEM application is authenticated by multiple signature file / certificate file pairs, the GEM terminal shall check all the signature file / certificate file pairs before deciding that a file fails authentication. It is dependent on receiver policy whether additional signature file / certificate file pairs are checked once one satisfactory pair has been found.

12.4.4 Integration

Logically a file is authenticated as follows:

a) Confirm that the file is listed in the hash file located in the same directory as the file to be authenticated.

b) Verify that the file contents and the corresponding digest value are consistent.

c) Recursively ascend the directory hierarchy checking that the hash file in each directory is authenticated by the hash file in its parent directory until a directory is found that contains one or more signature files.

   Such a directory is termed the root directory of an authenticated subtree.

NOTE 1: This means that there is no requirement to progress above the root of the authenticated subtree to examine further hash files. If such a directory has a digest type other than "Non authenticated" in its parent directory, this is only significant when verifying the hash file of the parent directory.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 198 of 861

NOTE 2: The presence of more than one signature file enables more than one set of organizations to authenticate a subtree.

NOTE 3: When authenticating a signed application, a GEM terminal can select any one of these certificates to use to authenticate all subsequent classes or files loaded.

Where a GEM terminal trusts more than one of these certificates, it should attempt to select the most trusted of these. Apart from this, the mechanism used to make this selection is implementation dependent.

d) For a signature file locate the corresponding certificate file (where the x portion of the signature file's file name identifies the certificate file to be used).

e) If the root certificate in the corresponding certificate file is unknown to the GEM receiver, return to step (d) and repeat for the other signature files.

f) Use the corresponding certificate file to verify that the signature correctly signs the hash file.

g) Verify that the attributes of the Subject field include content that matches the signalling information as per the requirements of clause 12.5.6.

h) Follow the certificate chain contained within the certificate file verifying each link in the chain until the link to the root certificate is found.

i) If the identified root certificate and all the intermediate certificates leading to it are "satisfactory", accept the files as being authenticated.

"Satisfactory" depends on the policies implemented in the receiver and other constraints expressed in the present document. In particular the requirements in clause 11.2.3, "Class Loading" for using the "same" leaf certificate to authenticate DVB-J class files shall be observed.

j) If the identified root certificate or any of the intermediate certificates leading to it are not "satisfactory", return to step (d) and repeat for the other signature files.

k) Dependent on receiver policy return to step (d) and repeat for other signature files.

NOTE 4: The above is a logical description of the process and does not constrain implementations to perform these steps in this exact order. E.g. hash files may be verified when descending an directory hierarchy rather than ascending one.

A file system may contain several independent authenticated subtrees, each tree with its own subtree root directory.

NOTE 5: For the method `org.dvb.dsmcc.DSMCCObject.getSigners`, the semantics defined in that method take precedence over these rules where there are conflicts.

12.5 Profile of X.509 certificates for authentication of applications

This clause identifies how ITU-T Recommendation X.509 [52] is profiled when used for authentication of broadcast GEM applications. This profile is a variation (in general a sub-set) of the internet profile defined in RFC 2459 [55]. This clause identifies the differences from the profile in RFC 2459 [55]. Clause 12.11, "The internet profile of X.509" summarizes the profile in RFC 2459 [55].

12.5.1 signatureAlgorithm

GEM supports 2 signature algorithms: MD5 with RSA and SHA-1 with RSA.

12.5.1.1 MD5 with RSA

The signature algorithm with MD5 and the RSA encryption algorithm is defined in RFC 2313 [53]. As defined in RFC 2313 [53], the ASN.1 OID used to identify this signature algorithm is:

```
md5WithRSAEncryption OBJECT IDENTIFIER ::= {
    iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 4 }
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 199 of 861

12.5.1.2 SHA-1 with RSA

The signature algorithm with SHA-1 and the RSA encryption algorithm is implemented using the padding and encoding conventions described in RFC 2313 [53]. The message digest is computed using the SHA-1 hash algorithm. The ASN.1 object identifier used to identify this signature algorithm is:

```
sha-1WithRSAEncryption OBJECT IDENTIFIER ::= {
    iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 5 }
```

12.5.1.3 parameters

For both of the two supported algorithms the parameters component shall be the ASN.1 type NULL.

12.5.2 signatureValue

The RSA signature generation process and the encoding of the result is described in detail in RFC 2313 [53].

12.5.3 version

The version field of the certificate shall signal v3. All implementations shall support the v3 extensions as required by clause 12.5.9, "Extensions".

12.5.4 issuer

12.5.4.1 minimum requirement

For GEM at least a Common Name attribute shall be provided. The text value of the attribute shall be non-empty. It shall be suitable for direct presentation to the user.

12.5.4.2 certificate authority responsibility

The senior certificate authority who signs a certificate shall oversee the attribute information to ensure that the information is suitable.

12.5.5 validity

The only allowed format for encoding time in the validity field is GeneralizedTime.

12.5.6 subject

The subject field is a "distinguished name". The following requirements are specified by GEM:

- The only allowed encoding attributes of the subject is `UTF8String` (see clause 12.4.3.4, "Name matching").
- The minimum set of attributes that shall be present in the subject are:
  - `commonName`
  - `countryName`
- If the certificate is a "leaf certificate" (see figure 14, "Certificate chain illustrated") then the subject shall also contain an `organizationName`.
- When encoded the `organizationName` carries organization specific text post fixed by the `organisation_id` of the authenticated files. This integer value is represented as a fixed length 8 character hexadecimal string (with leading zeros where required). So the `organizationName` takes the form:
  - `text + "." + organisation_id`

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Except for the presence of leading zeros, this is the same as clause 14.5, "Text encoding of application identifiers".

This field reproduces the `organisation_id` value from the application's application identifier, see clause 10.4.3, "Content of the Application Description". Before the application starts executing, the value of the `organisation_id` of at least one of the leaf certificates used in the authentication of the application's root subdirectory must match that in the application identifier. If this is not true, then the application is considered to have failed authentication. Applications which fail authentication at this point shall not be started on GEM terminals.

If the certificate is a "leaf certificate" then the subject may also contain up to four `organizationalUnitName` entries.

When encoded each `organizationalUnitName` entry carries a comma separated list of up to five original network identifiers each entry of which is a fixed length 5-character hexadecimal string encoded as follows:

- The first character is a network type identifier encoded as a single hexadecimal digit with one or more bits set and with the bits having the following meanings:

xxx1	terrestrial network
xx1x	cable network
x1xx	satellite network
1xxx	internet

- The next four characters are a hexadecimal representation, including leading zeros where necessary, of the original_network_id as defined in EN 300 468 [14].

Before the application starts executing, the network type and the original_network_id of the service carrying the application shall be matched against the entries in the comma separated lists in the `organizationalUnitName` entries in each of the leaf certificates used in the authentication of the application's root directory. The original_network_id of the service, shall be found by first looking in the transport_stream_loop of the NIT for the transport stream carrying the service and only if the service is not listed in that loop, looking in the SDT.

Certificates where none of the entries match shall not be used to authenticate applications. If no match is found in any of these leaf certificates then the application is considered to have failed authentication. Applications that fail authentication at this point shall not be started on GEM terminals.

NOTE: Where a particular original_network_id is uniquely used in a particular market, this feature enables certificate authorities to apply market specific policies when deciding which organisations are entitled to have certificates issued to them for that market. The extent to which this provides real additional real security depends on factors outside the scope of the present document.

12.5.7 SubjectPublic Key Info

GEM supports a single public key algorithm (RSA) for the subject public key.

The key lengths that implementation are required to support are addressed in annex G, "Minimum platform capabilities".

12.5.7.1 rsaEncryption

The OID rsaEncryption identifies RSA public keys:

```
rsaEncryption OBJECT IDENTIFIER ::= { pkcs-1 1}

pkcs-1 OBJECT IDENTIFIER  ::=  { iso(1) member-body(2) us(840)
    rsadsi(113549) pkcs(1) 1 }
```

The parameters field shall have ASN.1 type NULL.

12.5.7.2 subjectPublicKey

The RS subjectPublicKey BIT STRING shall be encoded using the ASN.1 type RSAPublicKey:

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 201 of 861

```
RSAPublicKey ::= SEQUENCE {
    modulus                                 INTEGER, -- n
    publicExponent                          INTEGER -- e -- }
```

The semantics of the modulus (n) and the public exponent (e) are defined in RFC 2313 [53].

12.5.8 Unique Identifiers

X.509 defines the `issuerUniqueID` and `subjectUniqueID` extensions.

CAs conforming to this profile shall not generate certificates with unique identifiers.

GEM terminals conforming to this profile are not required to be capable of parsing unique identifiers and making comparisons.

12.5.9 Extensions

The following restrictions and semantics are placed on the use of certificate extensions when used to authenticate applications.

Table 47: Profile for standard certificate extensions

Extension	In broadcasts	In implementations	Semantic
Authority key identifier	Opt.	Opt.	Shall not be marked critical
Subject key identifier	Opt.	Opt.	Shall not be marked critical
Key usage	Mand.	Mand.	If the keyUsage extension is marked critical, for the leaf certificate the bit digitalSignature shall be set, for other certificates the bit keyCertSign shall be set. If these bits are not set then the certificate shall be ignored by the implementation.
Private key usage period	Opt.	Opt.	Shall not be marked critical
Certificate policies	Opt.	Opt.	Shall not be marked critical
Policy mappings	Opt.	Opt.	Shall not be marked critical
Subject Alternative Name	Mand.	Opt.	Shall not be marked critical The subject name unambiguously identifies the subject. It is recommended that DVB GEM implementations can read rfc822Name (email address)
Issuer Alternative Name	Mand.	Opt.	Shall not be marked critical The issuer name unambiguously identifies the issuer. It is recommended that DVB GEM implementations can read rfc822Name (email address).
Subject Directory attributes	Opt.	Opt.	Shall not be marked critical
Basic Constraints	Opt.	Mand.	May be marked critical
Name Constraints	Opt.	Mand.	May be marked critical. DVB GEM decoders shall be able to recognize name constraints when GeneralName are either directoryName or rfc822 names.
Policy Constraints	Opt.	Opt.	Shall not be marked critical
Extended key usage field	Opt.	Opt.	Shall not be marked critical
CRL Distribution points	Opt.	Opt.	Shall not be marked critical

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 48: Key for table 47

Keyword	When applied to	
	broadcasts	**receivers**
Mandatory	Certificates shall include these extensions.	Receivers shall observe the semantic for this information when provided.
Optional	Certificates may or may be not include these extensions. The GEM specification does not define the use of these extensions and hence broadcasters should not use them.	Receivers can ignore these extensions if present.
Critical/not-critical	Broadcasters should not mark the receiver optional extensions as critical.	Certificates containing unrecognized critical extensions shall be considered as invalid. Receivers should recognize all the extensions that can be critical.

12.6 Security policy for applications

12.6.1 General principles

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

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

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

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

NOTE 1: Any local attributes on newly defined prefixed (qualified) XML element types that are used locally to those element types and are not intended to be used as global attributes should not be prefixed in accordance with the Per-Element-Type Partition defined by XML Namespaces [28], annex A.2, "XML Namespace Partitions".

This clause specifies the resource access policy for the downloaded applications. The resource access policy depends on two factors:

- The access rights requested by the broadcaster through the signalling.
- The access rights granted by the user.

The ultimate access rights that are granted to the applications are the intersection of the access rights requested by the broadcaster and the access rights granted by the user.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Unsigned applications have limited access to platform resources.

Unless specified elsewhere in the present document, signed applications have the same access rights as unsigned applications. An application broadcaster can request additional permissions to access specific resources by providing a signed "Permission Request File" along with the application. The syntax and semantics of the Permission Request File are defined in the following clauses. The permission request file may also contain a credential that indicates that a persistent file owned by another organization may be accessed. If the "Permission Request File" is not correctly authenticated the application is not granted any additional permissions but is not prevented from starting for this reason.

The way the user grants rights to the downloaded applications is implementation dependent and is not addressed by GEM.

For DVB-J applications, accessing a resource consists of method calls. Each method call that results in accessing the resource shall throw a security exception as defined in clause 11.10, "Java permissions" and the specifications of the java APIs concerned. For each resource subject to access restriction, the application can test whether it has been granted permissions to access it by using the corresponding java Permission class (see clause 11.8, "Security").

For a DVB-J application to be correctly authenticated, all the class files that the application consists of need to be signed, the signatures need to verify (see clause 12.4.4, "Integration") and the application_id needs to be from within the range allocated to signed applications (see table 13, "Value ranges for application_id"). If, during the loading or execution of the application the GEM detects a signed file containing a class that failed to pass the authentication process (e.g. because its actual hash value does not match the expected hash value), then the class shall be considered as not available.

When an application requests to retrieve data from a signed file, and the GEM terminal attempts to authenticate that file, and the authentication process fails, then the API concerned shall fail in a manner consistent with the defined behaviour of that API when the file exists but has no content in it.

NOTE 2: In order to be efficient, if a directory D contains objects that are likely to frequently change, it is advised to put a signature file in this directory D and to mark the directory D as non authenticated in the hashfile located in the parent directory of D. By doing so, it will limit the propagation of modifications to just one directory.

The authentication of a file is evaluated each time that the file is loaded from a transport connection. File version information in the transport system cannot be assumed to be secure.

Applications authors should be aware that deciding whether to grant a permission or not may, depending on the implementation, involve prompting and asking the end user. The latest point in time when the implementation must decide if an application has a permission or not is when the application either queries the presence of this permission for the first time or when it invokes an action that requires the permission for the first time. Application authors should be aware that in these situations, an implementation may prompt and ask the user. Depending on the implementation, this prompting (if necessary) can also happen at any point in time prior to this (e.g. at the application start up time).

A GEM terminal is required to be able to operate in a mode where it grants permission to provide access to all of the functionality required by the profiles and options that it supports when appropriately requested (e.g. via the permission request file). The mechanism for causing the terminal to operate in this mode is implementation-dependent. The granting of permissions for accessing functionality outside of the claimed GEM profile and options is not required.

NOTE 3: In the case of permissions represented by a Java class in DVB-J, this means that it will be possible to have such a permission granted if the corresponding class is required in the given profile, and it can be requested in the permission request file.

NOTE 4: This means that, for example, it is possible to grant the permission associated with dialling a phone number on a terminal that supports the interactive broadcast profile, even if the terminal implements the interaction channel using a cable modem. In this case, the dialling APIs will fail in a manner consistent with GEM.

12.6.2 Permission request file

12.6.2.0 General

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 204 of 861

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

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

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

NOTE 1: The policy for granting of permissions outlined in clause 12.6.1, "General principles" 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 [5] defines a "monitor application" that makes policy decisions. In cases such as this, the downloaded element is required to implement a policy consistent with GEM.

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

NOTE 3: If a terminal cannot support functionality implied by a permission tag, it still supports the syntax of the permission tag. E.g. `capermission` tag is supported, even if support for the GEM CA APIs is not present.

12.6.2.1 File encoding

12.6.2.1.1 XML

The Permission Request File is an XML File. Its syntax is defined by the following DTD. The `Name` used in the document type declaration shall be "`permissionrequestfile`".

12.6.2.1.2 MHP/GEM 1.0

The `PublicLiteral` to be used for specifying this DTD in document type declarations of the XML files is:

```
"-//DVB//DTD Permission Request File 1.0//EN"
```

and the URL for the SystemLiteral is:

```
"http://www.dvb.org/mhp/dtd/permissionrequestfile-1-0.dtd".
```

The DTD is:

```
<!ELEMENT permissionrequestfile
    (file?,capermission?,applifecyclecontrol?,returnchannel?,tuning?,
    servicesel?,userpreferences?,network?, dripfeed?, persistentfilecredential*)>
<!ATTLIST permissionrequestfile
 orgid CDATA #REQUIRED
 appid CDATA #REQUIRED
>

<!ELEMENT file EMPTY>
<!ATTLIST file
    value                           (true|false) "true"
>
<!ELEMENT capermission (casystemid)+>
<!ELEMENT casystemid EMPTY>
<!ATTLIST casystemid
    entitlementquery                             (true|false) "false"
    id                          CDATA #REQUIRED
    mmi                         (true|false) "false"
    messagepassing                           (true|false) "false"
    buy                         (true|false) "false"
>

<!ELEMENT applifecyclecontrol EMPTY>
<!ATTLIST applifecyclecontrol
    value                           (true|false) "true"
>

<!ELEMENT returnchannel (defaultisp?,phonenumber*)>
<!ELEMENT defaultisp EMPTY>
<!ELEMENT phonenumber (#PCDATA)>
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

```
<!ELEMENT tuning EMPTY>
<!ATTLIST tuning
    value                           (true|false) "true"
>

<!ELEMENT servicesel EMPTY>
<!ATTLIST servicesel
    value                           (true|false) "true"
>

<!ELEMENT userpreferences EMPTY>
<!ATTLIST userpreferences
    write                           (true|false) "false"
    read                            (true|false) "true"
>

<!ELEMENT network (host)+>
<!ELEMENT host (#PCDATA)>
<!ATTLIST host
    action                          CDATA #REQUIRED
>

<!ELEMENT dripfeed EMPTY>
<!ATTLIST dripfeed
    value                           (true|false) "true"
>

<!ELEMENT persistentfilecredential (grantoridentifier,expirationdate,filename+,signature,
certchainfileid)>
<!ELEMENT grantoridentifier EMPTY>
<!ATTLIST grantoridentifier
    id                          CDATA #REQUIRED
>
<!ELEMENT expirationdate EMPTY>
<!ATTLIST expirationdate
    date                            CDATA #REQUIRED
>
<!ELEMENT filename (#PCDATA)>
<!ATTLIST filename
    write                           (true|false) "true"
    read                            (true|false) "true"
>
<!ELEMENT signature (#PCDATA)>
<!ELEMENT certchainfileid (#PCDATA)>
```

12.6.2.1.3 MHP/GEM 1.1

The `PublicLiteral` to be used for specifying this DTD in document type declarations of the XML files is:

```
"-//DVB//DTD Permission Request File 1.1//EN"
```

and the URL for the SystemLiteral is:

```
"http://www.dvb.org/mhp/dtd/permissionrequestfile-1-1.dtd".
```

The DTD is:

```
<!ELEMENT permissionrequestfile
    (file?,capermission?,applifecyclecontrol?,returnchannel?,tuning?,
    servicesel?,userpreferences?,network?, dripfeed?, persistentfilecredential*,
    applicationstorage?,smartcardaccess?,provider?)>

<!ATTLIST permissionrequestfile
 orgid CDATA #REQUIRED
 appid CDATA #REQUIRED
>

<!ELEMENT file EMPTY>
<!ATTLIST file
    value                           (true|false) "true"
>

<!ELEMENT capermission (casystemid)+>
<!ELEMENT casystemid EMPTY>
<!ATTLIST casystemid
    entitlementquery                (true|false) "false"
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

```
    id                              CDATA #REQUIRED
    mmi                             (true|false) "false"
    messagepassing                  (true|false) "false"
    buy                             (true|false) "false"
>

<!ELEMENT applifecyclecontrol EMPTY>
<!ATTLIST applifecyclecontrol
    value                           (true|false) "true"
>

<!ELEMENT returnchannel (defaultisp?,phonenumber*)>
<!ELEMENT defaultisp EMPTY>
<!ELEMENT phonenumber (#PCDATA)>

<!ELEMENT tuning EMPTY>
<!ATTLIST tuning
    value                           (true|false) "true"
>

<!ELEMENT servicesel EMPTY>
<!ATTLIST servicesel
    value                           (true|false) "true"
>

<!ELEMENT userpreferences EMPTY>
<!ATTLIST userpreferences
    write                           (true|false) "false"
    read                            (true|false) "true"
>

<!ELEMENT network (host)+>
<!ELEMENT host (#PCDATA)>
<!ATTLIST host
    action                          CDATA #REQUIRED
>

<!ELEMENT dripfeed EMPTY>
<!ATTLIST dripfeed
    value                           (true|false) "true"
>

<!ELEMENT persistentfilecredential (grantoridentifier,expirationdate,filename+,signature,
certchainfileid)>
<!ELEMENT grantoridentifier EMPTY>
<!ATTLIST grantoridentifier
    id                              CDATA #REQUIRED
>
<!ELEMENT expirationdate EMPTY>
<!ATTLIST expirationdate
    date                            CDATA #REQUIRED
>
<!ELEMENT filename (#PCDATA)>
<!ATTLIST filename
    write                           (true|false) "true"
    read                            (true|false) "true"
>
<!ELEMENT signature (#PCDATA)>
<!ELEMENT certchainfileid (#PCDATA)>


<!-- ..................... new elements in MHP 1.1 ................ -->
<!ELEMENT applicationstorage (applicationstorageorg)*>
<!ATTLIST applicationstorage
    manageservice (true|false) "false"
    createservice (true|false) "false"
    deleteservice (true|false) "false"
    managecache (true|false) "false"
>

<!ELEMENT applicationstorageorg EMPTY>
<!ATTLIST applicationstorageorg
    orgid CDATA #REQUIRED
    storeservice (true|false) "false"
    removeservice (true|false) "false"
    storecache (true|false) "false"
    removecache (true|false) "false"
>
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 207 of 861

```
<!ELEMENT smartcardaccess EMPTY>

<!ELEMENT priviligedrce EMPTY>
<!ATTLIST privilegedrce
    value (internal|external) "internal"
>
<!ELEMENT provider (name)+ >
<!ELEMENT name (#PCDATA)>
<!ATTLIST name
    scope (application|system) "application"
>
```

12.6.2.1.4 MHP/GEM 1.2

The `PublicLiteral` to be used for specifying this DTD in document type declarations of the XML files is:

```
    "-//DVB//DTD Permission Request File 1.2//EN"
```

and the URL for the SystemLiteral is:

```
    "http://www.dvb.org/mhp/dtd/permissionrequestfile-1-2.dtd".
```

The DTD is:

```
<!ELEMENT permissionrequestfile
    (file?,capermission?,applifecyclecontrol?,returnchannel?,tuning?,
    servicesel?,userpreferences?,network?, dripfeed?, persistentfilecredential*,
    applicationstorage?,smartcardaccess?,provider?, servicetypepermission?,
    monitorapplication? )>

<!ATTLIST permissionrequestfile
 orgid CDATA #REQUIRED
 appid CDATA #REQUIRED
>

<!ELEMENT file EMPTY>
<!ATTLIST file
    value                           (true|false) "true"
>

<!ELEMENT capermission (casystemid)+>
<!ELEMENT casystemid EMPTY>
<!ATTLIST casystemid
    entitlementquery                            (true|false) "false"
    id                          CDATA #REQUIRED
    mmi                         (true|false) "false"
    messagepassing                          (true|false) "false"
    buy                         (true|false) "false"
>

<!ELEMENT applifecyclecontrol EMPTY>
<!ATTLIST applifecyclecontrol
    value                           (true|false) "true"
>

<!ELEMENT returnchannel (defaultisp?,phonenumber*)>
<!ELEMENT defaultisp EMPTY>
<!ELEMENT phonenumber (#PCDATA)>

<!ELEMENT tuning EMPTY>
<!ATTLIST tuning
    value                           (true|false) "true"
>

<!ELEMENT servicesel EMPTY>
<!ATTLIST servicesel
    value                           (true|false) "true"
>

<!ELEMENT userpreferences EMPTY>
<!ATTLIST userpreferences
    write                           (true|false) "false"
    read                            (true|false) "true"
>
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 208 of 861

```
<!ELEMENT network (host)+>
<!ELEMENT host (#PCDATA)>
<!ATTLIST host
    action                          CDATA #REQUIRED
>

<!ELEMENT dripfeed EMPTY>
<!ATTLIST dripfeed
    value                           (true|false) "true"
>

<!ELEMENT persistentfilecredential (grantoridentifier,expirationdate,filename+,signature,
certchainfileid)>
<!ELEMENT grantoridentifier EMPTY>
<!ATTLIST grantoridentifier
    id                          CDATA #REQUIRED
>
<!ELEMENT expirationdate EMPTY>
<!ATTLIST expirationdate
    date                            CDATA #REQUIRED
>
<!ELEMENT filename (#PCDATA)>
<!ATTLIST filename
    write                           (true|false) "true"
    read                            (true|false) "true"
>
<!ELEMENT signature (#PCDATA)>
<!ELEMENT certchainfileid (#PCDATA)>


<!-- ..................... new elements in MHP 1.1 ................ -->
<!ELEMENT applicationstorage (applicationstorageorg)*>
<!ATTLIST applicationstorage
    manageservice (true|false) "false"
    createservice (true|false) "false"
    deleteservice (true|false) "false"
    managecache (true|false) "false"
>

<!ELEMENT applicationstorageorg EMPTY>
<!ATTLIST applicationstorageorg
    orgid CDATA #REQUIRED
    storeservice (true|false) "false"
    removeservice (true|false) "false"
    storecache (true|false) "false"
    removecache (true|false) "false"
>

<!ELEMENT smartcardaccess EMPTY>

<!ELEMENT priviligedrce EMPTY>
<!ATTLIST privilegedrce
    value (internal|external) "internal"
>

<!ELEMENT provider (name)+ >

<!ELEMENT name (#PCDATA)>
<!ATTLIST name
    scope (application|system) "application"
>

<!-- ..................... new elements in MHP 1.2 ................ -->

<!ELEMENT servicetypepermission EMPTY>
    <!ATTLIST servicetypepermission
    type (broadcast | abstract.mso) "broadcast"
    action (own | all) "all"
    value (true | false) "false"
>

<!ELEMENT monitorapplication EMPTY>
<!ATTLIST monitorapplication
    value (true | false) "false"
    name (registrar | service | servicemanager | security | reboot | handler.appFilter |
systemevent)
    #IMPLIED
>
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 209 of 861

12.6.2.1.5 Number representation

12.6.2.1.5.1 Hex

Unless stated otherwise (e.g. clause 14.5, "Text encoding of application identifiers") each hexadecimal value has the following form "0x" Nhex where "N" specifies the fixed number of significant digits in the value and hex is specified by the following BNF:

```
hex = digit | "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f"
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

12.6.2.2 File integrity

If the permission file is not parsable as defined by the XML parsing rules (see clause 14.3, "XML notation") it shall be ignored and hence no additional permissions are granted (or not granted in the case of `SelectPermission`).

12.6.2.3 Example

```
<?xml version="1.0"?>
<!DOCTYPE permissionrequestfile PUBLIC "-//DVB//DTD Permission Request File 1.0//EN"
"http://www.dvb.org/mhp/dtd/permissionrequestfile-1-0.dtd">
<permissionrequestfile orgid="0x000023d2" appid="0x4020">

    <file value="true"></file>

    <capermission>
        <casystemid
            id="0x1111" messagepassing="true"
            entitlementquery="true" mmi="false">
        </casystemid>
    </capermission>

    <applifecyclecontrol value="true"></applifecyclecontrol>

    <returnchannel>
        <defaultisp></defaultisp>
        <phonenumber>+3583111111</phonenumber>
        <phonenumber>+3583111112</phonenumber>
        <phonenumber></phonenumber>
    </returnchannel>

    <tuning value="false"></tuning>

    <servicesel value="true"></servicesel>

    <userpreferences read="true" write="false"></userpreferences>

    <network>
        <host action="connect">hostname</host>
    </network>

    <persistentfilecredential>
        <grantoridentifier id="0x05"></grantoridentifier>
        <expirationdate date="24/12/2032"></expirationdate>
        <filename read="true" write="false">5/15/dir1/scores</filename>
        <filename read="true" write="false">5/15/dir1/names</filename>
        <signature>02320329329293293292149314392942394329423943 2
        </signature>
        <certchainfileid>3</certchainfileid>
    </persistentfilecredential>

    <provider>
        <name scope="application"> 00000B.EMV_PK11.VISA_REVOLVER </name>
    </provider>

</permissionrequestfile>
```

12.6.2.4 Permission request file name and location

The format for the permission request file name is:

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 210 of 861

```
'dvb.'<application name>'.perm'
```

The prefix "dvb" identifies this as a well known file specified by GEM. The portion "application name" carries the file name of the initial file of the application excluding any file name extension or suffix. The initial file depends on the application type as is shown in table 49 for the types defined in the present document.

Table 49: Application names for different application types

Application type		Path from which file name shall be extracted
Value	Meaning	
0x0001	DVB-J	The name initial_class_name, see MHP [1], table 102
0x0002	DVB-HTML	The name initial_path_bytes, see MHP [1], table 104

This file shall be located in the same directory as the initial file.

12.6.2.5 Permission Request file

12.6.2.5.1 Minimum permissions

If the permission file does not contain a valid permissionrequestfile element it shall be ignored and hence no additional permissions are granted (or not granted in the case of `SelectPermission`).

12.6.2.5.2 Syntax and semantics

```
<!ATTLIST permissionrequestfile
    orgid CDATA #REQUIRED
    appid CDATA #REQUIRED
>
```

`orgid`: This attribute is a hexadecimal string (hex_string) that conveys the organisation id of the associated application. The encoding of this value is specified in clause 14.5, "Text encoding of application identifiers".

`appid`: This attribute is a hexadecimal string (hex_string) that conveys the application id of the associated application. The encoding of this value is specified in clause 14.5, "Text encoding of application identifiers".

12.6.2.5.3 Defaults

NOTE: As defined by XML 1.0 [60], the defaults for attributes defined in the DTD (see clause 12.6.2.1, "File encoding") only apply when the element is present but the attribute concerned is omitted.

12.6.2.6 Credentials

Support for this mechanism is optional.

Since the "`signature`" and "`certchainfileid`" element documented in table 50 is dependent on security files used in the MHP application authentication mechanism, stated elements may be modified depending on the application authentication mechanism used. Thus, GEM terminal specifications that do not use the functional equivalent named "Application Authentication" as defined in clause 15.6, "Functional equivalents" shall define an equivalent mechanism using the same XML syntax.

NOTE: The credentials mechanism allows applications from different sources (e.g. signers) to share data. This particular mechanism is not required, but writers of GEM terminal specifications are invited to consider how such data sharing might be enabled in their network and/or environment. Possible mechanisms might involve file permissions, or a network-based solution.

A credential contains a resource description and is used to allow the owner of this resource (the grantor) to grant to the permission request file's application the right to access it. In GEM, the only resource that can be contained in a credential is a file (or a set of files of a directory). Credentials shall not grant access to directories, only files within them. This type of credential is named `persistentfilecredential` in the XML DTD. The credential contains an expiration date that allow the grantor to grant access to its resource for a limited duration. The credential is signed by the grantor. The signature checking is done by the implementation by getting the certificate of the grantor. The certificate can be found thanks to the information contained in the `certchainfileid` element.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The certificate file that leads to the public key of the signature shall be placed either in the same directory as the permission request file containing the credential or in one of its parent directories. The directory containing the permission request file shall be searched first and then recursively ascend the directory hierarchy checking certificate files until one is found with the file name matching the contents of the `certchainfileid` field of the credential. If the root of the file system is reached without finding a matching certificate then the file access shall not be granted.

The grantoridentifier in the `persistentfilecredential` shall match the `organisation_id` contained in the Subject organisationName field of the leaf certificate for file access to be granted. The grantoridentifier shall match the organisation id of the owner of the file to which access is granted by this `persistentfilecredential`. If either of these is untrue, access shall not be granted.

The persistent file credential is transmitted using the following XML DTD syntax:

```
<!ELEMENT persistentfilecredential (grantoridentifier,expirationdate,filename+,signature,
certchainfileid)>
<!ELEMENT grantoridentifier EMPTY>
<!ATTLIST grantoridentifier
    id CDATA #REQUIRED
>
<!ELEMENT expirationdate EMPTY>
<!ATTLIST expirationdate
    date CDATA #REQUIRED
>
<!ELEMENT filename (#PCDATA)>
<!ATTLIST filename
    write (true|false) "true"
    read (true|false) "true"
>
<!ELEMENT signature (#PCDATA)>
<!ELEMENT certchainfileid (#PCDATA)>
```

Table 50 provide more information about the different elements.

Table 50

Elements	Comments
`grantoridentifier`	This element contains the 32 bit organization id identifying the grantor organization. The encoding of the CDATA attribute id of this element is "0x" followed by the encoding defined in 14.5, "Text encoding of application identifiers".

`expirationdate`	This element contains the expiration date of this credential.The implementation should ignore the certificate if the date has expired. The `CDATA` attribute `date` shall follow the following BNF syntax:
	`2dec "/" 2dec "/" 4dec ; e.g. "01/12/2001"`
	
	where:
	`dec = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"`
	
	Where the first 2 digits are the day number in the month, the second two digits are the month number in the year and the last 4 digits are the year. All values are in accordance with the Gregorian calendar.
	NOTE: Numbers start counting from one and not zero unlike the java.util.GregorianCalendar class.

`filename`	This element contains the filename path and the read/write access rights that are granted on the file. The element consists of a `CDATA` string following the following BNF syntax:
	`filename = persistentfilename`
	`| persistentpath "/"`
	`persistentfilename`
	`| persistentpath "/" "*"`
	`| persistentstpath "/" "-"`
	
	where:
	"/" is the file separator
	"*" used at the end of a pathname indicates all files contained in that directory
	"-" used at the end of a pathname indicates all files contained in that directory and existing subdirectories recursively at the time

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Elements	Comments
	the permission request file is parsed. `persistentpath`, `persistentfilesubstring` and `persistentfilename` are defined in clause 14.6.1, "Persistent storage". The file path is relative to the path obtained from the `dvb.persistent.root property`. Receivers are required to support the same restrictions on the lengths of `persistentfilesubstrings` and `persistentfilesuffixes` as specified in clause 14.6.1, "Persistent storage". The file path shall not start with a "/". It is relative to the path obtained from the `dvb.persistent.root` property. The element has two attributes read and write that can take the value "true" or "false".
`signature`	This element contains a signature from the grantor. Signature structure is as defined in clause 12.4.2, "SignatureFile". The signature is encoded using the Base64 content transfer encoding as specified in section 6.8 of RFC 2045 [59].
`certchainfileid`	This element contains the identifier of the leaf certificate, i.e.the "x" in the file name "dvb.certificates.x".

The signature is computed on the binary concatenation of the following fields in the following order.

Table 51

Fields	Binary content
`grantee_identifier.organisation_id`	32 bits
`grantee_identifier.application_id`	16 bits
`grantor_identifier (organisation_id)`	32 bits
`expiration_date`	10 characters (e.g. "10/12/2001") in ASCII

filenames and actions (in the order they appear in the XML document) i.e.:
 	
`for (i=0;i<filenumber;i++) {`
`  read`
`  write`
`  filepath`
`}`
	
	`4 or 5 char ("true" or "false")`
	`4 or 5 char ("true" or "false")`
	`string in ASCII (without any string termination character)`

The implementation will check the validity of the credential by checking the signature with the grantor's public key that can be found in the grantor's certificate. Certificates are carried by the grantee in the file format defined clause 12.4.3, "CertificateFile". The certification chain authenticates the grantor's certificate. This chain shall derive from one of the root authorities.

The right to access a file shall not include the right to create it.

12.6.2.7 File Access

12.6.2.7.1 Unsigned applications

Unsigned applications have no access to the persistent storage.

12.6.2.7.2 Policy for signed applications

No access to the persistent storage, unless otherwise indicated in the Permission File.

Applications may request access to persistent storage either by credentials or the "file" element or both. These two mechanisms are independent. Once an application has access to a file granted by one mechanism, the other mechanism need not be considered further. Specifically, once a credential to access a file is granted to an application, the permissions for the file encapsulated by FileAccessPermission are no longer relevant.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

NOTE: The intended use-case for this is a file containing an end-user's credit card details which may be intended to only be accessed by applications with a credential, e.g. from a bank. For this use-case, the application that put the credit card details in the file needs to set all the FileAccessPermissions to false in order to achieve the desired result.

When the permission request file requests access to persistent storage and this is granted, the file access policy is derived from the policy used in the Unix world.

An application owns the files it has created. The root directory of the persistent file namespace is defined by the 'dvb.persistent.root' property as described in clause 11.5.6, "Persistent Storage API". The files owned by an application shall be located in sub-directories below this directory, specifically one of the following two choices:

a) The sub-directory whose name is the `organisation_id` of the application concerned.

b) A sub-directory of the immediately previously defined directory whose name is the application_id of the application concerned.

The encodings of the organisation identifier and application identifier are as defined in clause 14.5, "Text encoding of application identifiers". By default, files created by an application will have owner read/write access only.

An Application can modify the access rights to a file it owns as follows:

- It can grant a read-only access, a write-only access or a read-write access to all applications having the same organisationId value.
- It can grant a read-only access, a write-only access or a read-write access to all applications.
- Write access to a directory is required to add or remove an entry in a directory.
- Read access to a directory is required to list the contents of a directory or access any of the files contained within in it.
- To read the contents of a file or directory, permission to read that file or directory and all directories on the path to the root of the persistent file system is required.

An application shall be granted access to a file if it qualifies for such access by application, organization or world access permissions.

See org.dvb.io.persistent see annex K, "DVB-J persistent storage API".

12.6.2.7.3 Permission request syntax

```
<!ELEMENT file EMPTY>
<!ATTLIST file
 value (true|false) "true"
>
```

12.6.2.8 CA API

12.6.2.8.0 GEM Introduction

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

12.6.2.8.1 Unsigned applications

An unsigned application cannot access the following methods:

- `CAModule.buyEntitlements`

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- CAModule.openMessageSession
- CAModuleManager.addMMIListener
- CAModule.queryEntitltements
- CAModule.listEntitlements

12.6.2.8.2 Signed applications

By default, an application has limited access to the CA API functions (same default rights as an unsigned application).

In particular, the following method calls are not accessible to an unsigned application:

- CAModule.buyEntitlements
- CAModule.openMessageSession
- CAModuleManager.addMMIListener
- CAModule.queryEntitlements
- CAModule.listEntitlements

The permission request file requests the MHP terminal to grant additional rights to the application with the ConditionalAccess Permission described below.

12.6.2.8.3 Conditional Access Permission syntax

The ConditionalAccess Permission is optional. When not present, the application has the default access rights. When present, the permission request file overrides the default rights for this application.

```
<!ELEMENT capermission (casystemid)+>
<!ELEMENT casystemid EMPTY>
<!ATTLIST casystemid
    entitlementquery                          true|false) "false"
    id                          CDATA #REQUIRED
    mmi                         (true|false) "false"
    messagepassing                          (true|false) "false"
    buy                         (true|false) "false"
>
```

The string specifying the CA system IDs has the following syntax:

```
CAIds                   =   CASystemId | "[" CASystemId "-" CASystemId "]" | "*"
CASystemId              =   "0x" 4hex
```

See clause 12.6.2.1.5.1, "Hex".

12.6.2.9 Application lifecycle control policy

Applications shall not launch broadcast applications that are not signalled in the currently selected service for the service context in which the application wishing to do the launching is running.

12.6.2.9.1 Unsigned applications

An unsigned broadcast application can launch any application visible in the listing API that is signalled in the currently selected service for the service context in which the application wishing to do the launching is running.

An unsigned application can control (pause, stop, resume) the lifecycle of an application it has launched.

An unsigned application cannot control the lifecycle of an application it has not launched.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 215 of 861

12.6.2.9.2 Default policy for signed applications

By default, a signed application has the same rights as an unsigned application as concerns the application lifecycle control policy.

These default rights can be overridden by the permission request file as described below.

12.6.2.9.3 Syntax

```
<!ELEMENT applifecyclecontrol EMPTY>
<!ATTLIST applifecyclecontrol
 value (true|false) "true"
>
```

value: When the boolean value is set to `true`, this means that the application can control the lifecycle of all the applications signalled in the currently selected service for the service context in which the application wishing to do the launching is running. When set to `false` the policy is as in clause 12.6.2.9.2, "Default policy for signed applications".

12.6.2.10 Return channel access policy

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

12.6.2.10.1 Unsigned applications

An unsigned application may not use the return channel.

12.6.2.10.2 Signed applications

By default, a signed application may not access the return channel, unless otherwise specified by the permission request file. The syntax of the return channel permission is so that it describes the phone numbers that the application may try to dial.

12.6.2.10.3 Return channel permission syntax

```
<!ELEMENT returnchannel (defaultisp?,phonenumber*)>
<!ELEMENT phonenumber (#PCDATA)>
<!ELEMENT defaultisp EMPTY>
```

The syntax of the phone number string is as defined below.

```
phonenumber             = "+" digit digits | digits
digits                  = "" | digit digits
digit                   = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

The presence of the `defaultisp` tag indicates that the application is allowed to use the default ISP connection. If this tag is not present, then the application shall not be able to specify connecting to the default target except where the connection parameters for that target are explicitly specified by the application and are included in the phone number string.

When a phone number is given, this number defines a prefix of the allowed phone numbers and applications are allowed to call to all numbers that start with one of the prefixes defined in the permission request file, subject to the MHP terminal granting this right (see clause 12.6.1, "General principles").

NOTE: By defining an empty phone number tag (i.e. empty string as the prefix), the application could try to call to any phone number.

12.6.2.11 Tuning access policy

This clause discusses the permissions for tuning access. As discussed in 11.10, "Java permissions" of the present document, the class `TunerPermission` is not required to be present for GEM terminal specifications that do not support the "Tuning API" (clause 11.6.3) listed in table Error! Reference source not found. of the present document;

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 216 of 861

for such GEM terminal specifications the policies specified in clauses 12.6.2.11.1 and 12.6.2.11.2 do not apply. If the "Tuning" functional equivalent is not included, it is to be silently ignored. However, the syntax of the `tuning` tag as specified in clause 12.6.2.11.3, "Tuner Permission syntax" shall be supported in all cases; if the tuning API is not included, it is to be silently ignored.

12.6.2.11.1 Unsigned applications

An unsigned application may not tune using the Tuning API.

12.6.2.11.2 Signed applications

By default, a signed application may not tune using the Tuning API. However, the right to tune can be requested with the Tuning permission that can be put in the permission request file.

12.6.2.11.3 Tuner Permission syntax

The syntax of this permission is as follows:

```
<!ELEMENT tuning EMPTY><

<!ATTLIST tuning
 value (true|false) "true"
>
```

The value true requests the permission to tune using the Tuning API.

12.6.2.12 Service selection policy

12.6.2.12.1 Unsigned applications

Unsigned applications may not select a new service.

12.6.2.12.2 Signed applications

By default, signed applications can select any new service, unless otherwise specified in the permission request file.

Service selection might require return channel access. If an application attempts to do a service selection using a locator that refers to an AIT File delivered over the return channel, and a return channel has not already been established or the application does not have permission to establish a connection via the default isp (see clause 12.6.2.10, "Return channel access policy"), then the service selection shall fail in a manner consistent with the defined failure mode for service selection when a security violation is encountered (e.g. a DVB-J application using the service selection API shall be thrown a SecurityException).

NOTE: See also clause 12.14.1, "Permission for application loading from the return channel".

12.6.2.12.3 Service Selection Permission

The syntax of the service selection permission is as follows:

```
<!ELEMENT servicesel EMPTY>

<!ATTLIST servicesel
 value (true|false) "true"
>
```

If no service selection permission is present in the permission request file, the application has the default right, i.e. it can select any service.

The value "false" in this item in the permission request file, denies the right to select a new service.

12.6.2.13 Media API access policy

The media API is inside the sandbox.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12.6.2.14 Inter-application communication policy

12.6.2.14.1 Unsigned applications

Unsigned applications are allowed to communicate with each other through the inter-application communication API. However, unsigned and signed applications shall only be able to communicate with each-other using the inter-application communication API in the "dvb:/ixc/" namespace.

12.6.2.14.2 Signed applications

Signed applications signalled in the same service are allowed to communicate with each other through the interapplication communication API. Signed applications shall only be able to communicate with unsigned applications using the inter-application API in the "dvb:/ixc/" namespace.

12.6.2.15 User Setting and Preferences access policy

12.6.2.15.1 Unsigned applications

Read access to:

- User Language.
- Parental Rating.
- DefaultFontSize.
- Country Code.

An unsigned application cannot access (neither read nor write) other preferences.

12.6.2.15.2 Signed applications

By default, same as unsigned applications. The permission request file may include items that request read access to all user preferences and/or write access to all user preferences.

12.6.2.15.3 Permission syntax

```
<!ELEMENT userpreferences EMPTY>

<!ATTLIST userpreferences
 write (true|false) "false"
 read (true|false) "true"
>
```

True value in the write and read attribute means that the permission for writing and reading, respectively, is requested.

12.6.2.16 Network permissions

12.6.2.16.1 Unsigned applications

Unsigned applications can not have access to the return channel and therefore can not access remote network hosts.

12.6.2.16.2 Signed applications

For signed applications, the permission request file can contain a set of permissions that specify the hosts and actions for which permissions are requested.

12.6.2.16.3 Permission syntax

```
<!ELEMENT network (host)+>
<!ELEMENT host (#PCDATA)>
<!ATTLIST host
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

```
 action CDATA #REQUIRED
>
```

The two strings that specific the host and the action shall be formatted as specified in the `java.net.SocketPermission` class as defined in clause 11.8, "Security".

12.6.2.17 Dripfeed permissions

12.6.2.17.1 Unsigned applications

Has no access to drip feed mode.

12.6.2.17.2 Default policy for signed applications

No access to drip feed mode unless otherwise indicated in the Permission file. When an application is granted access to the drip feed mode, it is able to instantiate a `DripFeedDataSource`.

12.6.2.17.3 Permission request syntax

```
<!ELEMENT dripfeed EMPTY>
<!ATTLIST dripfeed
    value (true|false) "true"
>
```

12.6.2.18 Privileged Runtime Code Extension Permission

NOTE: DVB-J prohibits runtime code extension from data sources less secure than the original program code by virtue of the fact that interoperable DVB-J applications cannot directly create a ClassLoader, so this permission does not apply to DVB-J. See clause 11.10.1.4, "java.lang.RuntimePermission" and DVBClassLoader in annex I, "DVB-J fundamental classes".

12.6.2.18.1 Unsigned Applications

Unsigned applications shall be permitted to do runtime code extension with string data from any source.

12.6.2.18.2 Signed applications with no permission request file

Signed applications with no permission request file shall be permitted to do runtime code extension with string data from any source.

12.6.2.18.3 Signed applications with a permission request file

By default, a signed application with a permission request file is prohibited from doing any runtime code extension from a string. The permission request file may include items that request the ability to do runtime code extension with internal strings only, or with both internal an external strings.

12.6.2.18.4 Permission request syntax

```
<!ELEMENT priviligedrce EMPTY>
<!ATTLIST privilegedrce
    value (internal|external) "internal"
>
```

12.6.2.19 Application storage

12.6.2.19.1 Unsigned applications

Have no permission to store applications.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12.6.2.19.2 Default policy for signed applications

By default signed applications do not have permission to store any applications.

Permission to store applications can be requested using the permission request file.

12.6.2.19.3 Permission request syntax

```
<!ELEMENT applicationstorage (applicationstorageorg)*>
<!ATTLIST applicationstorage
    manageservice: (true|false) "false"
    createservice: (true|false) "false"
    deleteservice: (true|false) "false"
    managecache: (true|false) "false"
>
```

manageservice: When set to "true", requests a permission to manage a stored application service with the organization ID of the current application. If set to false, this permission is not requested.

createservice: When set to "true", requests a permission to create a stored application service with the organization ID of the current application. If set to false, this permission is not requested.

deleteservice: When set to "true", requests a permission to remove a stored application service with the organization ID of the current application. If set to false, this permission is not requested.

managecache: When set to "true", requests a permission to manage an application cache with the organization ID of the current application. If set to false, this permission is not requested.

```
<!ELEMENT applicationstorageorg EMPTY>
<!ATTLIST applicationstorageorg
    orgid CDATA #REQUIRED
    storeservice (true|false) "false"
    removeservice (true|false) "false"
    storecache (true|false) "false"
    removecache (true|false) "false"
>
```

orgid: The `organisation_id` of the organization whose application and stored services this permission request concerns. This field is encoded in hexadecimal form as specified in clause 14.5, "Text encoding of application identifiers". Alternatively, this field may be set to "*" to indicate all organization IDs.

storeservice: When set to "true", requests a permission to store applications of the organization identified by the `organisation_id` into a stored service. If set to false, this permission is not requested.

removeservice: When set to "true", requests a permission to remove applications of the organization identified by the `organisation_id` from a stored service. If set to false, this permission is not requested.

storecache: When set to "true", requests a permission to store applications of the organization identified by the `organisation_id` into a cache. If set to false, this permission is not requested.

removecache: When set to "true", requests a permission to remove applications of the organization identified by the `organisation_id` from a cache. If set to false, this permission is not requested.

12.6.2.20 Non-CA smart card access

12.6.2.20.1 Unsigned applications

Have no permission to access smart cards.

12.6.2.20.2 Default policy for signed applications

By default signed applications do not have permission to access smart cards.

Permission to access smart cards can be requested using the permission request file.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12.6.2.20.3 Permission request syntax

```
<!ELEMENT smartcardaccess EMPTY>
```

When the smartcardaccess element is included in the permission request file, the permission to access smart cards is requested. If the element is not included in the permission request file, this permission is not requested.

12.6.2.21 Provider Management

12.6.2.21.1 Unsigned applications

Have no permission to install or remove providers.

12.6.2.21.2 Signed applications

By default, signed applications have no permissions to add or remove providers.

Permission to insert or to remove a provider can be requested by requested using the permission request file.

An application shall only be granted permissions to insert or to remove providers that have the same organisation identifier as the application.

12.6.2.21.3 Permission request syntax

```
<!ELEMENT provider (name)+ >
<!ELEMENT name (#PCDATA)>
<!ATTLIST name
    scope (application|system) "application"
>
```

The provider name shall be prefixed with the GEM organisation identifier of the organisation which is in charge of it. The syntax for the GEM orgId shall be compliant with clause 12.6.2.1.5.1, "Hex".

The provider name can be "GEM_orgId.Provider_name.Card_name". For example "0x0000000B. EMV_PK11.VISA_REVOLVER".

12.6.2.22 Service type selection policy

12.6.2.22.1 Unsigned applications

Service type selection policy is irrelevant for unsigned applications since they are denied the ability to select a new service by clause 12.6.2.12.1, "Unsigned applications".

12.6.2.22.2 Signed applications

By default a signed application running in an AbstractService shall be granted a this permission of type "abstract.mso" and other signed applications shall be granted this permission with type of "broadcast".

NOTE: The type "broadcast" includes stand-alone stored services 9.6.2, "Stored services" and return channel signalled services 9.6.1, "Applications loaded from the interaction channel".

Entries in the permission request file may be used to add extra permissions or to over ride the default permission.

12.6.2.22.3 Permission request file syntax

```
<!ELEMENT servicetypepermission EMPTY>
<!ATTLIST servicetypepermission
    type (broadcast | abstract.mso) "broadcast"
    action (own | all) "all"
    value (true | false) "false"
>
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 221 of 861

12.6.2.23 Privileged application access

12.6.2.23.1 Unsigned applications

Unsigned applications shall not be able to access any privileged APIs.

12.6.2.23.2 Signed applications

Only applications successfully authenticated as defined in clause 12.18, "Authentication of privileged applications" shall be able to access privileged APIs.

Multiple instances of the "monitorapplication" element may appear, one for each type of access that is requested.

12.6.2.23.3 Permission request file syntax

```
<!ELEMENT monitorapplication EMPTY>
<!ATTLIST monitorapplication
    value (true | false) "false"
    name (registrar | service | servicemanager | security | reboot | handler.appFilter |
systemevent)
    #IMPLIED
>
```

12.7 Example of creating an application that can be authenticated

12.7.1 Scenario Example

This clause is informative and gives an example of how a file system carrying an application can be organized.

In this example, the file system carries single signed application Xlet1.

The main class of the application is the file `root/Xlet1/classes/Xlet1.class`, other files comprising the application are the following classes:

- `root/Xlet1/classes/foo1.class`
- `root/Xlet1/classes/subclasses/sub1.class`
- `root/Xlet1/classes/subclasses/sub2.class`

Xlet1 consumes data located in the file `root/Xlet1/data/Xlet1.dat`. This file is not authenticated.

Each subdirectory that contains signed files contains a hashfile.

Assumptions in this example:

- All the *.class files are signed.
- The file `root/Xlet1/data/Xlet1.dat` is not signed.
- The only digest algorithm used is MD5.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The file structure is shown below.

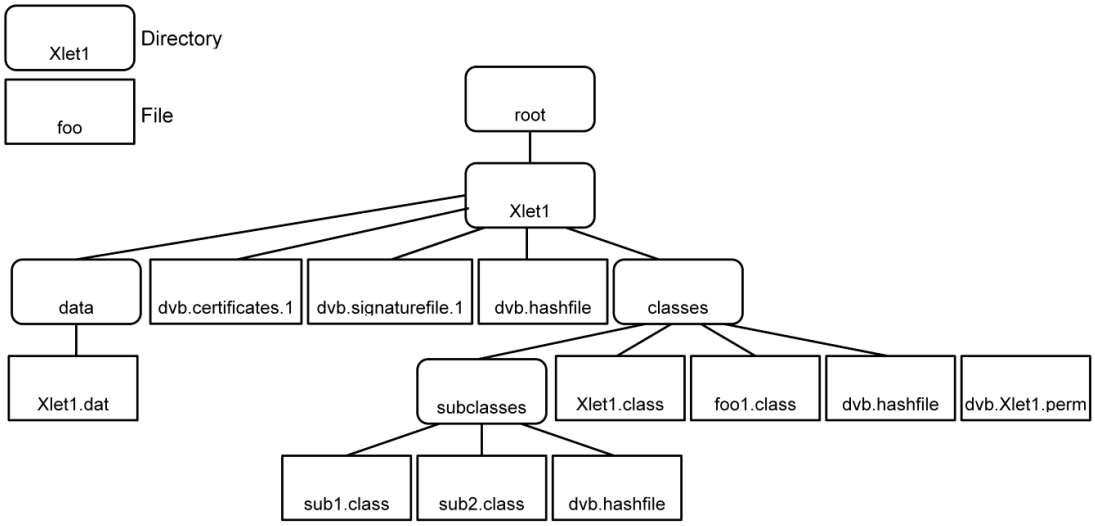


Figure 15: Example of an authenticated file tree

12.7.2 Hashes and signature computations:

12.7.2.1 Computation of the hashes of the root/Xlet1/classes/subclasses directory

a) Initialize the MD5 algorithm.

b) Apply the contents of the file "`root/Xlet1/classes/subclasses/sub1.class`" to the MD5 algorithm then apply the contents of the file "`root/Xlet1/classes/subclasses/sub2.class`" to the MD5 algorithm. The cumulative result from these two files is the result H0.

c) Construct the contents of the hash file for the directory `root/Xlet1/classes/subclasses/` as follows:

Table 52: root/Xlet1/classes/subclasses/dvb.hashfile

Field	Comment
1	One digest
1	Type of digest algorithm = MD5
2	Number of entries over which a MD5 hash has been computed
`sub1.class`	List of names of entries
`sub2.class`	
H0	MD5 hash of files `sub1.class` and `sub2.class`

NOTE: In this example we have chosen that the digests for this directory result from processing the concatenation of the contents of the files in the directory.

d) Create the hash file "`root/Xlet1/classes/subclasses/dvb.hashfile`" with the above contents.

12.7.2.2 Computation of the hashes of the of root/Xlet1/classes directory

e) Initialize the MD5 algorithm, compute the MD5 hash H2 using the contents of the file "`root/Xlet1/classes/subclasses/dvb.hashfile`".

f) Initialize the MD5 algorithm, compute the MD5 hash H3 using the contents of the file "`root/Xlet1/classes/Xlet1.class`".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

g) Initialize the MD5 algorithm, compute the MD5 hash H4 using the contents of the file "root/Xlet1/classes/foo1.class".

h) Initialise the MD5 algorithm, compute the MD5 hash H5 using the contents of the file "root/Xlet1/classes/dvb.Xlet1.perm".

i) Construct the contents of the hash file for the directory root/Xlet1/classes/ as follows:

Table 53: root/Xlet1/classes/dvb.hashfile

Field	Comment
3	Three digests
1	Type of digest algorithm = MD5
1	Number of entries over which a MD5 hash has been computed
subclasses	List of names of entries
H2	MD5 hash of subclasses directory
1	Type of digest algorithm = MD5
1	Number of entries over which a MD5 hash has been computed
Xlet1.class	List of names of entries
H3	MD5 hash of file Xlet1.class
1	Type of digest algorithm = MD5
1	Number of entries over which a MD5 hash has been computed
foo1.class	List of names of entries
H4	MD5 hash of file foo1.class
1	Type of digest algorithm = MD5
1	Number of entries over which a MD5 hash has been computed
dvb.Xlet1.perm	List of names of entries
H5	MD5 hash of file dvb.Xlet1.perm

NOTE: In this example we have chosen to individually hash each object in this directory.

j) Create the hash file "root/Xlet1/classes/dvb.hashfile" with the above contents.

12.7.2.3 Computation of the hashes of the of root/Xlet1 directory

k) Initialize the MD5 algorithm, compute the MD5 hash H5 using the contents of the file "root/Xlet1/classes/dvb.hashfile".

l) Construct the contents of the hash file for the directory root/Xlet1/classes/ as follows:

Table 54: root/Xlet1/dvb.hashfile

Field	Comment
3	Three digests
1	Type of digest algorithm = MD5
1	Number of entries over which a MD5 hash has been computed
classes	List of names of entries
H5	MD5 hash of the classes directory
0	Type of digest algorithm = non authenticated data
1	Number of entries over which a MD5 hash has been computed
data	List of names of entries
<no digest in this case>	
0	Type of digest algorithm = non authenticated data
1	Number of entries over which a MD5 hash has been computed
dvb.certificates.1 List	List of names of entries
<no digest in this case>	

NOTE: the root/Xlet1/classes entry is the only authenticated entry of the root/Xlet1 directory

m) Create the hash file "root/Xlet1/dvb.hashfile" with the above contents.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 224 of 861

12.7.2.4 Computation of the signature

n) Initialize the MD5 algorithm, compute the MD5 hash H6 using the contents of the file `root/Xlet1/dvb.hashfile`.

o) RSA-encrypt H6 with the private key corresponding to the public key that can be found in the leaf certificate in the file "`root/Xlet1/dvb.certificates.1`". In this example this has serial number 0123456.

p) ASN.1 encode the following structure:

- AuthorityCertIssuerName: Name of the CA.
- AuthorityCertSerialNumber: 0123456.
- HashSignatureAlgorithm: MD5.
- SignatureValue: result of step (o).

q) Put this structure into the signature file "`root/Xlet1/dvb.signaturefile.1`".

NOTE: The file system could contain other Xlets in their own authenticated sub-trees. If these use the same certificate chain as Xlet1 (above) then logically the certificates file is replicated at the root of each of the authenticated sub-trees. Some file systems support symbolic links. These may in some cases improve the efficiency of the broadcast.

12.8 Void

12.9 Certificate management

12.9.1 Certificate Revocation Lists

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

12.9.1.1 Introduction (informative)

Certificates may be revoked prior to their expiration time, e.g. if the broadcaster's private key is assumed to be compromised, or the broadcaster is no longer to be certified by the CA. Each CA publishes a list of revoked certificates, called a CRL (Certificate Revocation List). This contains the list of the serial number of revoked certificates.

During the validation process of a certificate chain, the CRL of each certification authority on the certification path is checked.

The number of CRLs to be supported per level is defined in annex G, "Minimum platform capabilities".

12.9.1.2 Distribution of CRLs (informative)

Two routes from the broadcaster to the GEM terminal can be envisaged:

- Via the return channel.
- In the broadcast MPEG stream.

12.9.1.2.1 Distribution via return channel

In the certificate extension fields, there is an optional field called "CRL Distribution points". This field can hold a URL pointing to a crlFile which can be downloaded.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

This approach is suitable for obtaining CRLs relating to TLS authentication of return channel exchanges. It is not suitable for delivering CRLs for broadcast application authentication as:

- Not all GEM profiles require return channel support.
- GEM terminals may be able to receive broadcast applications when not connected to a return channel.
- It would not be acceptable to require a return channel session to authenticate each broadcast application.

12.9.1.2.2 Distribution via MPEG stream

For a GEM terminal without a working return channel, the only way to deliver CRLs is via the broadcast MPEG Transport Stream.

12.9.1.3 CRL retention

12.9.1.3.1 Requirement

GEM terminals shall retain CRLs or the information they contain (including serial numbers) in persistent storage because:

- This enhances security as it defends against attacks with signals that filter out the CRL and use a revoked certificate.
- It is more efficient to cache CRLs rather than downloading the CRLs for authentication of each application.

12.9.1.3.2 Storage requirement

The minimum amount of persistent storage required to store CRLs is addressed in clause 12.12, "Platform minima".

12.9.1.3.3 Storage management

The broadcast CRL and RCMM signalling (see clause 12.9.2, "Root certificate management") manages the use of the persistent storage.

If the CRL of a non-root certificate CA becomes too large the CA's certificate itself can be revoked by its parent CA who adds it to their CRL. For root certificates, the RCMM mechanism can be used.

12.9.1.4 CRL file location and naming convention

For CRLs that are authenticated by a non-root certificate the format of the name of files carrying CRLs shall be:

```
"dvb.crl.x"
```

In this case the "x" portion of the filename corresponds to the "x" portion of a certificate filename for the certificate file that authenticates the CRL.

For CRLs that are authenticated by a root certificate the format of the name of files carrying CRLs shall be:

```
"dvb.crl.root.x"
```

In this case the "x" portion of the filename is just a discriminator to ensure non-collision of CRL file names in the event that there is more than one in this directory.

The CRL filename may not be constant through time or across broadcasts. So, implementations shall not rely on this filename when caching the CRL.

All CRL files shall be located in a subdirectory of the root of the file system called "dvb.crl". The location of certificate files authenticating the CRL files shall follow the same rules as for the location of certificates relative to a signature file. That is the certificate files shall either be in the dvb.crl directory or in the root directory. See clause 12.4.3.5, "CertificateFile location and naming conventions".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 226 of 861

12.9.1.5 Operational model

Receivers are expected to inspect the set of CRL files periodically and cache the revocation information for future use. This inspection process can assume that a file system will not normally carry certificates revoked by CRL files carried in the same file system. So, inspection of the CRL set in a file system is not a precondition to authenticating other files in that file system.

GEM does not address the reliability with which implementations are required to consider certificate revocations when authenticating files.

12.9.1.6 Examples

12.9.1.6.1 Revocation of a broadcaster's certificate

If the broadcaster B's certificate is compromised:

a) The certification authority CA01 adds the serial number of broadcaster B's certificate to its CRL.

b) CA01 then sends the new CRL file to the broadcasters that use CA01.

c) These broadcasters (broadcaster A for instance) broadcast the new CRL file.

As soon as a GEM terminal has downloaded the new CRL file (after selecting one of broadcaster A's channel), the GEM terminal's CRL cache in persistent storage is updated. The GEM terminal is then protected against any malicious usage of the compromised certificate.

To continue authenticating applications broadcast "B" will require a new certificate.

12.9.1.6.2 Revocation of a CA's certificate.

If the CRL of CA01 becomes too big, CA01's certificate could be revoked:

a) The root certification authority RCA0 adds the serial number of CA01's certificate to its CRL.

b) RCA0 sends the new CRL file to the broadcasters that use RCA0.

c) These broadcasters broadcast the new CRL file.

As soon as a GEM terminal has downloaded the new CRL file, its CRL cache in persistent storage for RCA0 is updated and CA01's CRL is removed from the cache (as CA01 has been revoked).

12.9.1.7 CRL format

Each CRL file contains the CRL of one certification authority.

The encoding of the CRL is defined in the ITU-T Recommendation X.509 [52] is reproduced below for information. The fields Version, Time, CertificateSerialNumber correspond to fields with the same names in the certificate see clause 12.11, "The internet profile of X.509":

```
CertificateList ::= SEQUENCE {
        tbsCertList                     TBSCertList,
        signatureAlgorithm              AlgorithmIdentifier,
        signatureValue BIT STRING }

TBSCertList ::= SEQUENCE {
        version                         Version OPTIONAL,
                                        -- if present, shall be v2
        signature                       AlgorithmIdentifier,
        issuer                          Name,
        thisUpdate                      Time,
        nextUpdate                      Time OPTIONAL,
        revokedCertificates             SEQUENCE OF SEQUENCE {
        userCertificate                 CertificateSerialNumber,
        revocationDate                  Time,
        crlEntryExtensions              Extensions OPTIONAL
                                        -- if present, shall be v2
        } OPTIONAL,
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 227 of 861

```
        crlExtensions                   [0] EXPLICIT Extensions OPTIONAL
                                        -- if present, shall be v2
}
```

12.9.1.8 Profile of CRL

The profile of fields that correspond to fields in the certificate follow the profile in clause 12.5, "Profile of X.509 certificates for authentication of applications". This applies to the following fields:

signatureAlgorithm : follows clause 12.5.1, "signatureAlgorithm".

signatureValue: follows clause 12.5.2, "signatureValue".

version: follows clause 12.5.3, "version".

signature: follows clause 12.5.1, "signatureAlgorithm".

issuer: follows clause 12.5.4, "issuer".

thisUpdate: Publication date of this CRL. Follows the encoding of Time used for validity. See clause 12.5.5, "validity".

nextUpdate: Publication date of the next version of the CRL. Follows the encoding of Time used for validity. See clause 12.5.5, "validity".

userCertificate: SerialNumber of the revoked certificate. It is used to identify the revoked certificate.

The serial number shall be unique for a given CA. So, the pair [issuerName, serialNumber] shall be unique.

revocationDate: Date of revocation for a given certificate. Follows the encoding of Time used for validity. See clause 12.5.5, "validity".

crlExtensions: The syntax of the Extensions element is reproduced in clause 12.11.1.13, "Extensions". This element allows multiple extension elements to be carried. The set of extensions defined for certificates and CRLs is reproduced at clause 12.11.2, "Standard certificate extensions".

The following crlExtension shall be supported:

- AuthorityKeyIdentifier as defined in clause 12.4.2.1.

  This extension identifies the certificate that is needed to check the signature of the CRL.

  The crlExtension `AuthorityKeyIdentifier` shall be included in every CRL. GEM implementations shall use the `AuthorityKeyIdentifier` to find the certificate that carries the public key that is used to check the signature of the CRL and ignore any CRLs where this check fails or where the AuthorityKeyIdentifier is not present.

Other crlExtensions are optional.

12.9.1.9 CRL Processing

CRLs, or the information they contain (including serial numbers), shall be kept in persistent storage in the GEM terminal.

When a GEM terminal finds a file with the file name format given in clause 12.9.1.4, "CRL file location and naming convention", it shall do the following sequence:

a) Get the field thisUpdate and compare it with the last update for the CRL. If thisUpdate is not after the last update, ignore the CRL.

b) The signature of the CRL is verified and the signing certificate is verified to be from a trusted source. If the certificate cannot be trusted, the CRL is ignored. For CRLs which are authenticated by a non-root certificate, the certificate shall be trusted only if the chain from that certificate to the root in its certificate file is verified and the root certificate is one of those resident in the GEM receiver. For CRLs which are authenticated by a root certificate, the certificates which can be trusted shall be the root certificates resident in the GEM receiver.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

c) Process the content of the CRL message:

- Store the list of serial numbers of revoked certificates in persistent storage.
- Update the stored value of thisUpdate with the value from the CRL.

d) If a CA's certificate has been revoked, remove its CRL (or the information which that CRL contained) if it was stored in the GEM terminal.

During the validation process of a certificate chain, the CRL of each certification authority on the certification path is checked.

12.9.2 Root certificate management

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

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

12.9.2.1 Introduction

Clause 12.9.2 specifies an MHP root certificate management protocol that fulfils the requirements of GEM, but another protocol syntax and another distribution means are possible. Broadly speaking, GEM places requirements on both the semantics of a root certificate management message and requirements underlying its processing. GEM does not, however, mandate the syntax of the protocol message and its distribution means; this is left for GEM terminal specifications to define, either by using the MHP definition or by defining a functional equivalent.

Every compliant MHP terminal will have to maintain a set of X.509 root certificates in persistent storage. These root certificates will be placed in the MHP terminal by its manufacturer during the manufacturing process.

It could be necessary to update the set of root certificates for MHP terminals that are already deployed. Possible reasons that could require such an update include:

- A root certificate becoming compromised.
- Technical developments (such as the emergence of factorization algorithms) that require the use of greater key lengths in root certificates to provide adequate security.
- Retirement of a certificate due to its age.

So, It is necessary to have a standard mechanism to update this set.

NOTE: A manufacturer specific mechanism could require a different message for each manufacturer and so would be more expensive in terms of broadcast bandwidth.

The mechanism specified here uses messages called RCMM (Root Certificate Management Messages). These messages contain a set of new root certificates to add and a reference to the root certificates to remove.

12.9.2.2 Security of RCMM

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

RCMM are authenticated by multiple signatures. An RCMM message will be accepted by an MHP terminal if and only if it has at least N signatures.

The initial value of N and the maximum value of N that MHP terminals will ever be asked to support are specified in clause 12.12, "Platform minima".

The use of multiple signatures guarantees that the set of root certificates can be updated securely even if one of the root certificates has been compromised.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

RCMM can update the number of signatures required for future RCMM using the nextNbOfSignature field.

The RCMM message shall be signed with the key of the certificates to be removed.

12.9.2.3 Format of RCMM

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

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

Table 55: Root Certificate Management Message description

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

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

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

thisUpdate: Date of issue of the message.

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

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

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

12.9.2.4 Distribution of RCMM

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

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

12.9.2.5 RCMM Processing

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

When an MHP terminal finds RCMM messages in a root directory, it shall perform the following sequence of operations:

a) If there has never been any RCMM processing (last update to be stored in the receiver not initialized), then process sequentially according to steps (b) to (j) all `dvb.rcmm.<x>` messages in ascending order including the `dvb.rcmm` message as the final step. Otherwise, identify the `dvb.rcmm.<x>` message whose field thisUpdate is the closest after the last update and process sequentially according to steps (b) to (j) all following `dvb.rcmm.<x>` in ascending order including the `dvb.rcmm` message as the final step. If the field thisUpdate from the `dvb.rcmm` message is not after the last update, ignore the messages.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 230 of 861

For each RCMM message identified in step (a), perform the following:

b) If there is a nextNbOfSignatures in the RCMM, and if nextNbOfSignatures will be greater than the number of remaining root CAs (i.e. the number of root CA that would remain if the RCMM was processed), ignore the RCMM.

c) Get the number signatures from the RCMM, if this number is lower than the minimum required, ignore the message.

d) If the RCMM contains references to root certificates to remove, check this RCMM is signed with the keys belonging to the certificates to be removed. If at least one of these signatures is missing, ignore the RCMM message.

e) Check all the signatures of the RCMM. If any of the signatures is invalid, ignore the message.

f) Process the content of the RCMM message, add and remove root certificates according to the RCMM message.

g) Store in persistent storage the date of thisUpdate as the date of the last update.

h) If a root certificate is removed, the CRLs associated are also removed.

i) If there is a nextNbOfSignatures in the RCMM, replace the minimum number of signature required by the nextNbOfSignatures.

j) Since this process will permanently modify the terminal behaviour and is highly sensitive for security, this is allowed to have an implementation dependent enabling / confirmation step prior to performing the action (e.g. it may prompt the end user for verification). Such an enabling / confirmation step is allowed to reject this RCMM action.

The implementation shall ensure the following:

- The integrity of the persistent storage shall be kept if the power supply fails for any reason during the processing of any RCMM message in a `dvb.rcmm.<x>` file or the `dvb.rcmm` file. i.e. If the power is switched off during the processing of the RCMM message, the set of root certificates shall remain as if processing of that RCMM message had not started.

  I.e. iIf the power is switched off during the processing of the RCMM message, the set of root certificates shall remain as if processing of the RCMM message had not started.

- The minimum amount of persistent storage reserved to store the list of root certificate is specified in clause 12.12, "Platform minima". If there is not enough persistent storage available to process an RCMM message, it shall be ignored (to prevent inconsistencies).

NOTE: GEM is intentionally silent about the impact of RCMM processing on MHP applications which are already running.

12.9.2.6 Example: Renewal of a root certificate

Let's assume the root certification authority RCA has two certificates in each MHP termninal:RC0 and RC1. If RC0 is compromised, RCA may wish to renew this certificate using the following steps:

a) RCA generates a new key pair and a new self signed certificate RC2.

b) RCA provides new certificates signed by RC1 to all the entities authenticated by RCA.

c) Wait until all entities authenticated by RCA have switched to the new certificates and have stopped using RC0.

d) RCA generates an RCMM message to add RC2 and to remove RC0. This RCMM will be double signed by RC0 and RC1 keys.

e) RCA will delivers this RCMM message to the broadcasters to update the MHP terminals.

(The broadcasting period should be long enough to update almost all of the MHP terminals in the field.)

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

f) RCA will provide RC1 and RC2 to set top box manufacturers as the new list of root certificates to put in set top boxes.

12.9.3 Test certificates

There shall be a means to make test root certificates (as required by clause 12.12, "Platform minima") available while the terminal is being tested. The method for doing so shall be implementation dependent; it may involve either replacing non-test root certificates, or storing test root certificates in addition to non-test root certificates.

Test root certificates shall only be functional when the terminal is being tested. Objects signed with test root certificates shall not be used other than for the purposes of testing.

GEM is intentionally silent on the test root certificates to be used.

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

12.10 Security on the return channel

General purpose security for the return channel is provided by the TLS (Transport Layer Security) protocol as described in RFC 2246 [58].

12.10.1 GEM application functionality

When implementing return channel security a GEM application shall:

- Implement the cipher suites identified in clause 12.10.2, "TLS cipher suites".

A GEM application is not required to implement the following:

- The functionality of being a server for the TLS protocol.
- Compliance with SSL 3.0.

GEM terminals shall support client authentication for TLS where the source of keys is provided to the class `javax.net.ssl.SSLContext` via a `javax.net.ssl.KeyManager`.

12.10.2 TLS cipher suites

The minimum set of cypher tools that implementations of the GEM profile of TLS shall implement are:

- RSA ().
- MD5.
- SHA-1.
- DES.
- AES.

More detail of this requirement is given in table 56 (see RFC 2246 [58] for definition of the terms) which identifies which methods are required in a GEM implementation. TLS cipher suites using AES are defined in RFC 3268 [79].

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 56: Profile of cipher suites that implementations are required to support

CipherSuite	Key Exchange	Cipher	Hash	Value (hex)	GEM status
TLS_NULL_WITH_NULL_NULL (note)	NULL	NULL	NULL	00, 00	Required
TLS_RSA_WITH_NULL_MD5	RSA	NULL	MD5	00, 01	Required
TLS_RSA_WITH_NULL_SHA	RSA	NULL	SHA-1	00, 02	Required
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA	RSA_EXPORT	DES40_CBC	SHA-1	00, 08	Required
TLS_RSA_WITH_DES_CBC_SHA	RSA	DES_CBC	SHA-1	00, 09	Required
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES_EDE_CBC	SHA-1	00, 0A	Required
TLS_RSA_WITH_AES_128_CBC_SHA	RSA	AES	SHA-1	00, 2F	Required
TLS_RSA_WITH_AES_256_CBC_SHA	RSA	AES	SHA-1	00, 35	Required
NOTE: This cipher suite is only used by a TLS implementation during the negotiation of a connection. It is not required to be enabled as a cipher suite that is available for a negotiated connection.					

NOTE: The ciphers TLS_RSA_WITH_NULL_MD5 and TLS_RSA_WITH_NULL_SHA provide integrity checking but without confidentiality (i.e. data are in clear). Data encryption is very time consuming for a server. They are useful for applications in which data don't need to be encrypted but in which data integrity is very important. For these applications, the server will only have to compute a HMAC for every message exchanged.

12.10.3 Downloading of certificates for TLS

12.10.3.1 Introduction

Before the TLS connection can be established, a GEM application has to ensure that the certificate list sent by a server contains at least one trusted certificate. In computer environment, this is simply done by checking the list of certificates against one certificate that is resident in the computer.

In the GEM environment, a downloadable application can establish a TLS session. This can be used for e.g. sensitive transactions. In such a scenario, the application knows which server to connect to, and also knows one certificate against which it can check that a given certificate chain contains the expected certificate that it knows and trusts.

The API that is used by a downloadable application is described in clause 11.8.2, "APIs for return channel security". The process of server authentication involves the checking of the certificate chain sent by the TLS server.

This clause specifies how the GEM terminal identifies and manages the TLS certificates that are downloaded along with the application and how verification (or otherwise) is presented to the application).

12.10.3.2 Usage of certificate in TLS

12.10.3.2.1 When certificates are delivered with the application

One or several TLS root certificates can be optionally broadcast along with the application.

When the certificate chain sent by the TLS server is not compatible with any of the TLS root certificates sent with the application an `IOException` will be thrown.

12.10.3.2.1.1 Certificate file naming and location

To facilitate certificate chain checking the name of the certificate file shall be:

```
dvb.tls.organisation_id.application_id.x
```

where:

- "x" is an optional string discriminating certificates where necessary.
- The encoding of `organisation_id` and application_id are specified in clause 14.5, "Text encoding of application identifiers".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Location of the TLS certificates is application type dependent. See table 57.

Table 57: TLS certificate locator semantics

Application_type	Description
0x0000	Reserved.
0x0001	For DVB-J the TLS certificate(s) are placed in the base directory of the application as defined in MHP [1] clause 10.9.2 "DVB-J application location descriptor".
0x0002	For DVB-HTML the TLS certificate(s) are placed at the physical root of the application as defined in MHP [1] clause 10.10.2 "DVB-HTML application location descriptor".
0x0003 to 0xFFFF	Reserved for future use.

12.10.3.2.1.2 Certificate authentication

To be considered valid TLS certificates the certificate files shall be authenticated members of the same authenticated sub-tree as the application.

12.10.3.2.1.3 Certificate file format

Each TLS certificate file contains a single certificate. The file format is the same as that used by certificate files for application authentication, see clause 12.4.3, "CertificateFile".

12.10.3.2.2 When no certificates are provided

When there are no TLS certificates sent with the application then the implementation will allow connection to be established to any server. The application can then use the JSSE API (see clause 11.8.2, "APIs for return channel security") to retrieve the certificate chain and check that it contains what the application requires. In such a case both name and public keys need to be checked by the application if the application wants to be sure of the remote server.

12.10.3.2.3 CRL distribution points

GEM implementations are free to ignore this field during TLS server authentication over the return channel.

12.11 The internet profile of X.509 (informative)

The text that follows summarizes the technical features of RFC 2459 [55] and references the different profile decisions made for the different GEM application areas.

12.11.1 Main part of the certificate

12.11.1.1 Certificate

```
Certificate  ::=  SEQUENCE  {
       tbsCertificate          TBSCertificate,
       signatureAlgorithm      AlgorithmIdentifier,
       signatureValue          BIT STRING }
```

12.11.1.2 signatureAlgorithm

The signatureAlgorithm field contains the identifier for the cryptographic algorithm used by the CA to sign this certificate.

An algorithm identifier is defined by the following ASN.1 structure:

```
AlgorithmIdentifier  ::=  SEQUENCE {
                  algorithm           OBJECT IDENTIFIER,
                  parameters          ANY DEFINED BY algorithm OPTIONAL }
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The algorithm identifier is used to identify a cryptographic algorithm. The OBJECT IDENTIFIER component identifies the algorithm. The contents of the optional parameters field will vary according to the algorithm identified.

This field must contain the same algorithm identifier as the signature field in the sequence TBSCertificate.

See clause 12.5.1, "signatureAlgorithm".

NOTE: RFC 2459 [55], section 7.2 lists the signature algorithms supported by that profile.

For all of the currently specified algorithms possible non-NULL parameters shall be ignored.

12.11.1.3 signatureValue

The signatureValue field contains a digital signature computed upon the ASN.1 DER encoded tbsCertificate. The ASN.1 DER encoded tbsCertificate is used as the input to the signature function. This signature value is then ASN.1 encoded as a BIT STRING and included in the Certificate's signature field.

NOTE: RFC 2459 [55], section 7.2 describes in detail this process for the algorithms supported by that profile.

By generating this signature, a CA certifies the validity of the information in the tbsCertificate field. In particular, the CA certifies the binding between the public key material and the subject of the certificate.

See clause 12.5.2, "signatureValue".

12.11.1.4 tbsCertificate

The field contains the names of the subject and issuer, a public key associated with the subject, a validity period, and other associated information. The tbscertificate may also include extensions.

The pair issuer / serialNumber uniquely identifies the certificate.

```
TBSCertificate ::=  SEQUENCE  {
        version         [0]  EXPLICIT Version DEFAULT v1,
        serialNumber         CertificateSerialNumber,
        signature            AlgorithmIdentifier,
        issuer               Name,
        validity             Validity,
        subject              Name,
        subjectPublicKeyInfo SubjectPublicKeyInfo,
        issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version shall be v2 or v3
        subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version shall be v2 or v3
        extensions      [3]  EXPLICIT Extensions OPTIONAL
                             -- If present, version shall be v3 }
```

12.11.1.5 version

This field describes the version of the encoded certificate. When extensions are used, as expected in this profile, use X.509 version 3 (value is 2). If no extensions are present, but a UniqueIdentifier is present, use version 2 (value is 1). If only basic fields are present, use version 1 (the value is omitted from the certificate as the default value).

Implementations should be prepared to accept any version certificate. At a minimum, conforming implementations must recognize version 3 certificates.

```
Version ::=  INTEGER { v1(0), v2(1), v3(2) }
```

Generation of version 2 certificates is not expected by implementations based on this profile.

See clause 12.5.3, "version".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12.11.1.6 serialNumber

The serial number is an integer assigned by the CA to each certificate. It must be unique for each certificate issued by a given CA (i.e. the issuer name and serial number identify a unique certificate).

```
CertificateSerialNumber ::=  INTEGER
```

12.11.1.7 signature

This field contains the algorithm identifier for the algorithm used by the CA to sign the certificate.

This field must contain the same algorithm identifier as the signatureAlgorithm field in the sequence Certificate. The contents of the optional parameters field will vary according to the algorithm identified.

If the signatureAlgorithm is different from the algorithm identifier in the signature field, the signature shall be rejected as being inconsistent.

See clause 12.5.1, "signatureAlgorithm".

NOTE: RFC 2459 [55], section 7.2 lists the supported signature algorithms for that profile.

12.11.1.8 issuer

The issuer field identifies the entity who has signed and issued the certificate. The issuer field must contain a non-empty distinguished name (DN). The issuer field is defined as the X.501 type Name. ITU-T Recommendation X.501 [51] Name is defined by the following ASN.1 structures:

```
Name ::= CHOICE {
            RDNSequence }

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::=
            SET OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE {
            type        AttributeType,
            value       AttributeValue }

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType

DirectoryString ::= CHOICE {
            teletexString                   TeletexString (SIZE (1..MAX)),
            printableString                 PrintableString (SIZE (1..MAX)),
            universalString                 UniversalString (SIZE (1..MAX)),
            utf8String                      UTF8String (SIZE (1.. MAX)),
            bmpString                       BMPString (SIZE (1..MAX)) }
```

The Name describes a hierarchical name composed of attributes, such as country name, and corresponding values, such as US. The type of the component AttributeValue is determined by the AttributeType; in general it will be a DirectoryString.

See clause 12.5.4, "issuer".

12.11.1.9 validity

The certificate validity period is the time interval during which the CA warrants that it will maintain information about the status of the certificate. The field is represented as a SEQUENCE of two dates: the date on which the certificate validity period begins notBefore) and the date on which the certificate validity period ends (notAfter). Both notBefore and notAfter may be encoded as UTCTime or GeneralizedTime.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

CAs conforming to this profile must always encode certificate validity dates through the year 2049 as UTCTime; certificate validity dates in 2050 or later must be encoded as GeneralizedTime.

```
Validity ::= SEQUENCE {
        notBefore       Time,
        notAfter        Time }

Time ::= CHOICE {
        utcTime         UTCTime,
        generalTime     GeneralizedTime }
```

A subject certificate's validity is limited by the validity period of the issuer's certificate.

12.11.1.9.1 UTCTime

The universal time type, UTCTime, is a standard ASN.1 type intended for representation of dates and time. UTCTime specifies the year through the two low order digits and time is specified to the precision of one minute or one second. UTCTime includes either Z (for Zulu, or Greenwich Mean Time) or a time differential.

For the purposes of this profile, UTCTime values must be expressed Greenwich Mean Time (Zulu) and must include seconds (i.e. times are YYMMDDHHMMSSZ), even where the number of seconds is zero. Conforming systems must interpret the year field (YY) as follows:

- Where YY is greater than or equal to 50, the year shall be interpreted as 19YY; and
- Where YY is less than 50, the year shall be interpreted as 20YY.

12.11.1.9.2 GeneralizedTime

The generalized time type, GeneralizedTime, is a standard ASN.1 type for variable precision representation of time. Optionally, the GeneralizedTime field can include a representation of the time differential between local and Greenwich Mean Time.

For the purposes of this profile, GeneralizedTime values must be expressed Greenwich Mean Time (Zulu) and must include seconds (i.e. times are YYYYMMDDHHMMSSZ), even where the number of seconds is zero. GeneralizedTime values must not include fractional seconds.

See clause 12.5.5, "validity".

12.11.1.10 subject

The subject field identifies the entity associated with the public key stored in the SubjectPublicKeyInfo field. It thus represents the entity whose public key is certified. It is encoded as a Distinguished Name (see clause 12.11.1.8, "issuer"). This name must be unique for each subject entity certified by one CA as defined by the issuer field.

12.11.1.10.1 issuerUniqueID

This field is optional and appears in X.509 v2 or v3 only. It enables to reuse the IssuerName over time. It is redundant with the issuer Name, and it is proposed here that this field be not parsed by the client.

12.11.1.10.2 subjectUniqueID

This field is optional and appears in X.509 v2 or v3 only. It enables to reuse the subjectName over time. It is redundant with the subject Name, and it is proposed here not to use this field.

The subject name may be carried in the subject field and/or the subjectAltName extension. If the subject is a CA (e.g. the basic constraints extension, as discussed in RFC 2459 [55], section 4.2.1.10, is present and the value of cA is TRUE,) then the subject field must be populated with a non-empty distinguished name matching the contents of the issuer field (see RFC 2459 [55], section 4.1.2.4) in all certificates issued by the subject CA. If subject naming information is present only in the subjectAltName extension (e.g. a key bound only to an email address or URI), then the subject name must be an empty sequence and the subjectAltName extension must be critical.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Where it is non-empty, the subject field must contain an X.500 Distinguished Name (DN). The DN must be unique for each subject entity certified by the one CA as defined by the issuer name field. A CA may issue more than one certificate with the same DN to the same subject entity.

The subject name field is defined as the X.501 type Name. Implementation requirements for this field are those defined for the issuer field (see RFC 2459 [55], section 4.1.2.4). When encoding attribute values of type DirectoryString, the encoding rules for the issuer field must be implemented. Implementations of GEM must be prepared to receive subject names containing the attribute types required for the issuer field. Implementations of the present document should be prepared to receive subject names containing the recommended attribute types for the issuer field. The syntax and associated object identifiers (OIDs) for these attribute types are provided in the ASN.1 modules in RFC 2459 [55] Appendices A and B. Implementations of the present document may use these comparison rules to process unfamiliar attribute types (i.e. for name chaining). This allows implementations to process certificates with unfamiliar attributes in the subject name.

In addition, legacy implementations exist where an RFC 822 [26] name is embedded in the subject distinguished name as an EmailAddress attribute. The attribute value for EmailAddress is of type IA5String to permit inclusion of the character '@', which is not part of the PrintableString character set. EmailAddress attribute values are not case sensitive (e.g. "fanfeedback@redsox.com" is the same as "FANFEEDBACK@REDSOX.COM").

Conforming implementations generating new certificates with electronic mail addresses must use the rfc822Name in the subject alternative name field (see RFC 2459 [55], section 4.2.1.7) to describe such identities. Simultaneous inclusion of the EmailAddress attribute in the subject distinguished name to support legacy implementations is deprecated but permitted.

12.11.1.11 SubjectPublic Key Info

This field is used to carry the public key which is certified and identifies the algorithm with which the key is used. The algorithm is identified using the AlgorithmIdentifier structure (see clause 12.11.1.2, "signatureAlgorithm") and the public key is represented as a bitstring.

```
SubjectPublicKeyInfo  ::=  SEQUENCE  {
        algorithm            AlgorithmIdentifier,
        subjectPublicKey     BIT STRING  }
```

See clause 12.5.7, "SubjectPublic Key Info".

NOTE: RFC 2459 [55], section 7.3 lists the supported algorithms for that profile.

12.11.1.12 Unique Identifiers

These fields may only appear if the version is 2 or 3. The subject and issuer unique identifiers are present in the certificate to handle the possibility of reuse of subject and/or issuer names over time. This profile recommends that names not be reused for different entities and that Internet certificates not make use of unique identifiers. CAs conforming to this profile should not generate certificates with unique identifiers. Applications conforming to this profile should be capable of parsing unique identifiers and making comparisons.

```
UniqueIdentifier  ::=  BIT STRING
```

See clause 12.5.8, "Unique Identifiers".

12.11.1.13 Extensions

This field may only appear if the version is 3 and is optional. If present, this field is a SEQUENCE of one or more certificate extensions.

```
Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension

Extension  ::=  SEQUENCE  {
        extnID      OBJECT IDENTIFIER,
        critical    BOOLEAN DEFAULT FALSE,
        extnValue   OCTET STRING  }

The extensions are defined in ITU-T Recommendation X.509 [52].
```

See clause 12.5.9, "Extensions".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 238 of 861

12.11.2 Standard certificate extensions

See table 47, "Profile for standard certificate extensions".

12.11.2.1 Authority key identifier

This extension is used where an issuer has multiple signing keys. It provides a means of finding the public key that can be used to check the signature of the certificate.

The identification can be based on either the keyIdentifier or the on the pair (authorityCertIssuer, AuthorityCertSerialNumber). It is recommended to use the pair (authorityCertIssuer, AuthorityCertSerialNumber) instead of the keyIdentifier because there is no common agreement on the way to compute a unique KeyIdentifier.

```
AuthorityKeyIdentifier ::= SEQUENCE {
  keyIdentifier             [0] KeyIdentifier           OPTIONAL,
  authorityCertIssuer       [1] GeneralNames            OPTIONAL,
  authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }
```

12.11.2.2 Subject key identifier

This extension provides of identifying certificates that contain a particular public key.

```
SubjectKeyIdentifier ::= KeyIdentifier
```

12.11.2.3 Key usage

The key usage extension defines the purpose of the key contained in the certificate.

```
keyUsage ::= BIT STRING {
  digitalSignature      (0),
  nonRepudiation        (1),
  keyEncipherment       (2),
  dataEncipherment      (3),
  keyAgreement          (4),
  keyCertSign           (5),
  cRLSign               (6),
  encipherOnly          (7),
  decipherOnly          (8)
}
```

12.11.2.4 Private key usage period

This field is used to defined a period of validity for the private key which is different from the period of validity of the certificate. This is only meaningful for digital signature keys.

```
privateKeyUsagePeriod EXTENSION ::= {
  SYNTAX        PrivateKeyUsagePeriod
  IDENTIFIED BY id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
  notBefore        [0]  GeneralizedTime OPTIONAL,
  notAfter         [1]  GeneralizedTime OPTIONAL }
  ( WITH COMPONENTS {..., notBefore PRESENT} |
    WITH COMPONENTS {..., notAfter PRESENT} )
```

12.11.2.5 Certificate policies

This field is used to define a policy defining the purpose for which the certificate may be used.

Applications may have a list of specific policies they will accept, the certificate validation software must be able to compare the policy OIDs found in the certificate to that list.

```
certificatePolicies EXTENSION ::= {
  SYNTAX      CertificatePoliciesSyntax
  IDENTIFIED BY id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

```
PolicyInformation ::= SEQUENCE {
  policyIdentifier   CertPolicyId,
  policyQualifiers   SEQUENCE SIZE (1..MAX) OF
                     PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
  policyQualifierId          CERT-POLICY-QUALIFIER.&id
                             ({SupportedPolicyQualifiers}),
  qualifier                  CERT-POLICY-QUALIFIER.&Qualifier
          ({SupportedPolicyQualifiers}{@policyQualifierId})
          OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }
```

12.11.2.6 Policy mappings

This field is used to define equivalence between policies from different CA's policy Domain.

```
policyMappings EXTENSION ::= {
  SYNTAX      PolicyMappingsSyntax
  IDENTIFIED BY id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
  issuerDomainPolicy    CertPolicyId,
  subjectDomainPolicy   CertPolicyId }
```

12.11.2.7 Subject Alternative Name

This field is used to define additional identities for the subject name of the certificate (such as an internet email address).

```
subjectAltName EXTENSION ::= {
  SYNTAX    GeneralNames
  IDENTIFIED BY id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
  otherName                 [0]  INSTANCE OF OTHER-NAME,
  rfc822Name                [1]  IA5String,
  dNSName                   [2]  IA5String,
  x400Address               [3]  ORAddress,
  directoryName             [4]  Name,
  ediPartyName              [5]  EDIPartyName,
  uniformResourceIdentifier [6]  IA5String,
  iPAddress                 [7]  OCTET STRING,
  registeredID              [8]  OBJECT IDENTIFIER }
```

12.11.2.8 Issuer Alternative Name

This field is similar to the previous one for the issuer identity.

```
issuerAltName EXTENSION ::= {
  SYNTAX      GeneralNames
  IDENTIFIED BY id-ce-issuerAltName }
```

12.11.2.9 Subject Directory attributes

This field is used to carry optional directory attributes associated with the subject.

```
subjectDirectoryAttributes EXTENSION ::= {
  SYNTAX      AttributesSyntax
  IDENTIFIED BY id-ce-subjectDirectoryAttributes }

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 240 of 861

12.11.2.10 Basic Constraints

This field indicates if the subject of the certificate is a certification authority and how deep a certification path may exist through that path.

This extension shall appear in every CA Certificates. It will be used to prevent unauthorized entities to act as a CA.

```
basicConstraints EXTENSION ::= {
  SYNTAX      BasicConstraintsSyntax
  IDENTIFIED BY id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
  cA                  BOOLEAN DEFAULT FALSE,
  pathLenConstraint    INTEGER (0..MAX) OPTIONAL }
```

12.11.2.11 Name Constraints

This field indicates a namespace within which all subject names in subsequent certificates in a certification path shall be located.

```
nameConstraints EXTENSION ::= {
  SYNTAX      NameConstraintsSyntax
  IDENTIFIED BY id-ce-nameConstraints }

NameConstraintsSyntax ::= SEQUENCE {
  permittedSubtrees          [0]  GeneralSubtrees OPTIONAL,
  excludedSubtrees           [1]  GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
  base                GeneralName,
  minimum        [0]  BaseDistance DEFAULT 0,
  maximum        [1]  BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)
```

12.11.2.12 Policy Constraints

This field is only used in CA's certificate (i.e. not in end entity certificate). It is used to prohibit policy mapping or to require that each certificate in a path contain an acceptable policy identifier.

```
policyConstraints EXTENSION ::= {
  SYNTAX      PolicyConstraintsSyntax
  IDENTIFIED BY id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
  requireExplicitPolicy   [0] SkipCerts OPTIONAL,
  inhibitPolicyMapping    [1] SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)
```

12.11.2.13 Extended key usage field

This field is an extensions of the previous field keyUsage. It is used to define more purposes for which the certified public key may be used.

```
extKeyUsage EXTENSION ::= {
  SYNTAX      SEQUENCE SIZE (1..MAX) OF KeyPurposeId
  IDENTIFIED BY id-ce-extKeyUsage }

KeyPurposeId ::= OBJECT IDENTIFIER
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 241 of 861

12.11.2.14 CRL Distribution points

This field defines how CRL (Certificate revocation list) information can be obtained.

```
cRLDistributionPoints EXTENSION ::= {
  SYNTAX         CRLDistPointsSyntax
  IDENTIFIED BY  id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
  distributionPoint  [0]  DistributionPointName OPTIONAL,
  reasons            [1]  ReasonFlags OPTIONAL,
  cRLIssuer          [2]  GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
  fullName                  [0]  GeneralNames,
  nameRelativeToCRLIssuer   [1]  RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
  unused               (0),
  keyCompromise        (1),
  cACompromise         (2),
  affiliationChanged   (3),
  superseded           (4),
  cessationOfOperation (5),
  certificateHold      (6) }
```

12.12 Platform minima

For GEM terminal specifications that do not adopt the functional equivalent named "application authentication" as defined in clause 15.6, "Functional equivalents", GEM terminal specifications may specify different Platform minima than what is described in this clause.

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

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

GEM platform hardware is required to support the following minimum sizes to support the GEM security model:

- A value of 2 for N in clause 12.9.2.2, "Security of RCMM".
- A minimum depth of 5 levels in the certificate chain.
- A minimum of 8 CRLs.
- A minimum of 10 entries per CRL.
- A minimum of 3 root certificates (regardless of the number of underlying root certificate authorities).
- A minimum of 3 root certificates for testing (see clause 12.9.3, "Test certificates").

12.13 Plug-ins

The security aspects for the plug-in are the same as for any other DVB-J application. The security aspects of the delegated application cannot exceed the permissions available to the plug-in.

So, the plug-in will normally request the set of permissions required to support a delegated application. For a delegated application that follows the security model defined in the present document, the plug-in shall implement a security system to grant a sub-set of these permissions to the delegated application (see annex AF, "Plug-in APIs"). For a delegated application whose security model does not follow the present document, the plug-in can optionally implement a private security system to manage a sub-set of these permissions to the delegated application.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

12.14 Applications loaded from an interaction channel

Same as for applications delivered over the broadcast channel except where specified otherwise below.

12.14.1 Permission for application loading from the return channel

The following policy covers loading of applications over the interaction channel. It includes both initial loading to start and application and dynamic loading during the lifetime of an application. It covers both the code of an application and supporting data however loaded.

- The user enters a locator specifying the location of the AIT File into the navigator:

  In this case the user implicitly gives permission for the connection to retrieve the AIT File when they supply the locator.

  Applications launched by this mechanism inherit limited rights to use the return channel. Where the protocol ID is "3" the application gets rights to access the files specified by the transport protocol descriptor.

  Additional permissions can be granted by the normal permissions mechanism.

- A service has a broadcast Application Description and the transport protocol for the application is type "3" so ALL files are provided by the interaction channel.

  If the application launch is initiated by another application using the listing and launching API and it is necessary to open the interaction channel before the permissions of the launched application are evident then the permissions of the launching application are considered.

  If the application launch is initiated from service selection (autostart application on a service) then the behaviour depends on policy in the GEM terminal.

NOTE: This behaviour may, for example, be controlled by a terminal user preference.

- A service has a broadcast Application Description but a proportion of the files forming the application are provided via the interaction channel.

  If the permission request file and the authentication infrastructure are available without opening the interaction channel then these can be used to determine if opening the interaction channel is suitably authorized.

- An application selects a service using a locator pointing to an AIT File delivered over the return channel to perform service selection

  In this case the service selection succeeds only if the application has permission to do a service selection and the appropriate return channel access permission, as described in clause 12.6.2.12.2, "Signed applications".

If the permission request file is not available then, as above, the behaviour depends on policy in the GEM terminal.

12.15 Stored applications

At time of execution of an application, all files in stored applications must have been fully authenticated as defined by clause 12.4.4, "Integration". The extent to which this authentication is performed at time of storage and time of execution differs as described below. In all cases, the certificates used must be valid at the time of execution (e.g. they have not been expired or revoked, and the root certificates they use have not been removed).

The normal constraints on authentication, including clause 11.2.3, "Class Loading", shall be respected when executing an application. (This applies even if some class files are stored but others are not.) This may lead to some stored class files failing authentication even if all certificates are still valid.

Files that fail authentication shall be treated in the same manner as files that fail authentication from a carousel. If a stored file fails authentication the implementation shall not fall back to loading that file from carousel.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 243 of 861

12.15.1 Stand-alone stored applications

At time of application storage the stored files must be authenticated as if they were being loaded by the application. At this stage the terminal shall not treat class files specially.

In the case that a certificate that was validated during the store process has been revoked or has expired, any additional certificate chains that were not validated at storage time shall now be fully validated using the same rules as would apply to such a file from broadcast - i.e. clause 12.4.3.6, "Authentication rules".

If stand-alone stored applications access broadcast files (e.g. using the org.dvb.dsmcc API), the full requirements of clause 12.4.4, "Integration" shall apply to those files.

At time of execution, it shall be checked that any certificates used to authenticate credentials have neither expired nor been revoked.

12.15.2 Cached applications

It is implementation dependent how much authentication is performed at the time of storage.

NOTE 1: As an optimisation, implementations may calculate hash values of files at the time they are stored.

NOTE 2: Cached applications which will run without access to an object carousel for the stored files (e.g. not_launchable_from_broadcast set to '1') may be authenticated as described for stand-alone stored applications. The time at which the various steps of the authentication process are performed is not visible to applications.

12.16 Void

12.17 Authentication of unbound applications

All unbound applications shall be signed and successfully authenticated. Unsigned applications or signed applications that fail authentication shall not be started as unbound applications, even if they have been successfully installed as such.

NOTE: This functionality is not mandatory in any profile of GEM.

12.18 Authentication of privileged applications

Privileged applications shall be subject to additional authentication beyond the normal authentication mechanisms. GEM intentionally does not require any specific additional authentication mechanisms.

NOTE 1: System software download may be used in two ways. Privileged applications may be distributed as part of a system software download and authenticated as part of the authentication of that download.

NOTE 2: Another possible additional authentication mechanism could be to require such applications to be signed by an additional certificate whose availability is very tightly controlled. See clause 14.2.2.2.2, "Signed applications" of OCAP [5] for an example of such a mechanism. Such additional certificates could be distributed as part of authenticated system software download.

NOTE 3: Additional mechanisms (e.g. not involving system software download) may be defined in future versions of GEM or by other specifications extending GEM.

NOTE 4: This functionality is not mandatory in any profile of GEM.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 244 of 861

13 Graphics reference model

13.1 Introduction

The current document provides tools to control the positioning of: video on an output device, interface components such as buttons and lists, as well as raw graphical primitives.

Each screen connected to a GEM has three planes which are, from back to front, a background plane, a video plane and a graphics plane.

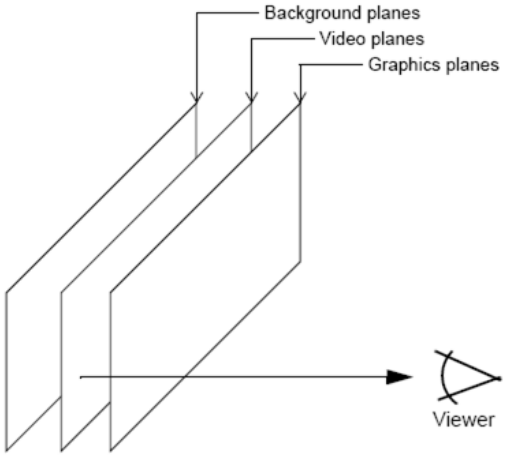


Figure 16: Illustration of the different types of display planes

The behaviour of the subtitle plane varies between implementations. API facilities are provided to allow applications to make the behaviour predictable. See clause 13.5, "Subtitles".

An application is provided with a contiguous rectangular region of the graphics plane in which it can draw (see clause 13.3.3, "HAVi devices and AWT components"). An application can place video, interface elements and graphics inside its rectangle on the graphic plane.

An application can also control video outside of the AWT hierarchy on the video plane, and place still images, video drips or solid colour in the background plane.

The GEM specification enables terminals to support multiple applications at any one time, each of which can have a sub area of the screen to which it can draw. The specification enables the areas to overlap. However, the minimum required support for these features is profile specific. See annex G, "Minimum platform capabilities".

NOTE: The mapping between planes in this reference model and planes in the hardware used in a GEM terminal is implementation dependent. In particular, it is certainly allowed to use planes which the terminal hardware considers to be "video" as what GEM considers to be "background" for the purposes of displaying MPEG stills and video drips in what GEM considers to be the background.

13.1.1 Interapplication interaction

If the presentations of different applications overlap, the areas obscured by other applications are clipped. Therefore where an application is translucent it will be blended with the video or background image behind it rather than being blended with another application.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

13.2 General Issues

13.2.1 Coordinate Spaces

The present document includes a number of coordinate systems for different purposes and includes the means to transform between these as needed:

- Input video space.

  This considers post upsampling MPEG pixels.

- Device space.

  Logical pixels in the various display devices. There may be different device spaces for the various device types (e.g. video and graphics).

- Normalized screen space.

  Normalized coordinates relative to the output (`HScreen`).

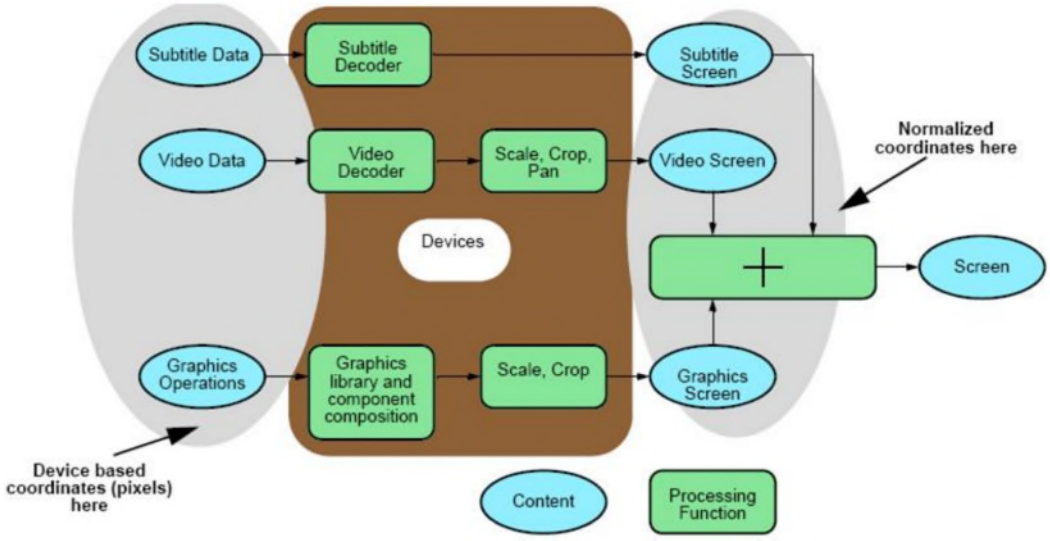


Figure 17

Various different interfaces provide access to different parts of the graphics and video systems using these coordinate spaces.

13.2.1.1 Normalized screen space

A normalized screen coordinate system supports references to positions and sizes in the video output from a GEM device without reference to any form of pixels.

This coordinate system describes the top left corner of the screen as {0, 0} and the bottom right corner of the screen as {1, 1}. This coordinate system is used in the positioning of graphics and video on the screen through the a number of classes in the `org.havi.ui` package and the JMF control `org.dvb.media.VideoPresentationControl`.

The normalized coordinate system is given through the `HScreen`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

13.2.1.2 User space

The coordinate space for graphics used in java.awt is defined by applications through the creation of an `HGraphicsDevice`. The root container, an instance of the class `org.havi.ui.HScene` can be placed within the normalized coordinate system. The `HSceneTemplate` class allows applications to express requirements for their top level container. Instances of this class are used by `HSceneFactory` to return instances of an `HScene`.

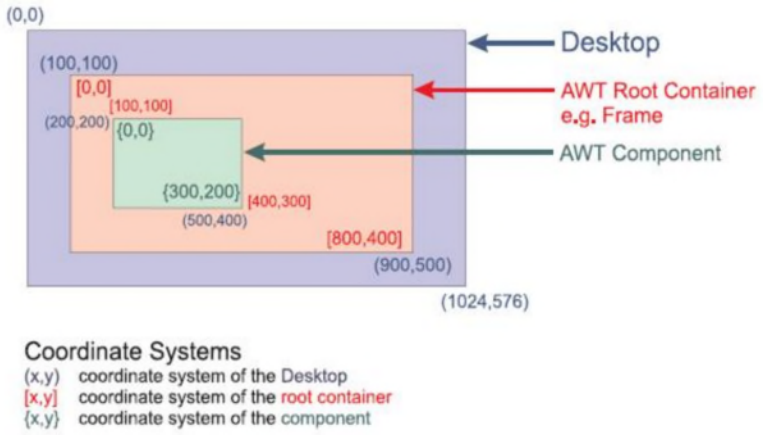


Figure 18: AWT Coordinate system in a computer environment

Figure 18 shows the AWT coordinate system in a normal computer environment. In a GEM device the `HGraphicsDevice` can be thought of being the desktop and the `HScene` as being the AWT RootContainer. Thus an `HScene` is placed within the coordinate system of the `HGraphicsDevice`. An `HScene` itself defines a new coordinate system which pixels are aligned to those of the `HGraphicsDevice` but the origin is translated (see figure 18).

The mapping between the user space and the normalized coordinate system is done given the size and location of the `HGraphicsDevice` in the normalized coordinate system and the resolution of the `HGraphicsDevice` in pixel.

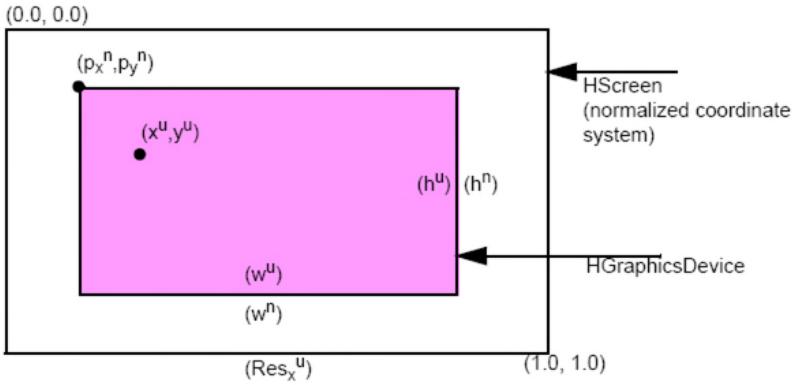


Figure 19: Conversion between normalized and user coordinate systems

(x^n, y^n) = normalized coordinate system of `HScreen`.

(x^u, y^u) = coordinate system of the `HGraphicsDevice` in pixels.

x^{vu} = "virtual" user coordinate system in pixels relative to the `HScreen`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Assume the HGraphicsDevice has the origin of (p_x^n, p_y^n) with a dimension of (w^n, h^n) and has a pixel resolution of w^u x h^u than the point (x^u, y^u) in normalized coordinates is given by:

$$x^n = \frac{x^{vu}}{\mathrm{Res}_x{}^{vu}} \qquad y^n = \frac{y^{vu}}{\mathrm{Res}_y{}^{vu}}$$

where:

$$\mathrm{Res}_x{}^{vu} = \frac{w^u}{w^n} \qquad \mathrm{Res}_y{}^{vu} = \frac{h^u}{h^n}$$

is the virtual resolution of the HScreen in (sub)pixels and:

$$x^{vu} = p_x{}^{vu} + x^u \qquad y^{vu} = p_y{}^{vu} + y^u$$

is the point (x, y) in the virtual coordinate space with:

$$p_x{}^{vu} = p_x{}^n \cdot \mathrm{Res}_x{}^{vu} \qquad p_y{}^{vu} = p_y{}^n \cdot \mathrm{Res}_y{}^{vu}$$

being the location of the HGraphicsDevice in the virtual coordinate system.

Given the Point (x^n, y^n) the point (x^u, y^u) is given by:

$$x^u = floor[(x^n - p_x{}^n) \cdot \mathrm{Res}_x{}^{vu} + 0.5]$$

and

$$y^u = floor[(y^n - p_y{}^n) \cdot \mathrm{Res}_y{}^{vu} + 0.5]$$

Given a HGraphicsDevice which is full screen this simplifies to:

$$\mathrm{Res}_x{}^{vu} = w^u \quad \text{and} \quad \mathrm{Res}_y{}^{vu} = h^u \quad , \quad x^{vu} = x^u \quad \text{and} \quad y^{vu} = y^u$$

thus:

$$x^n = \frac{x^u}{w^u} \qquad y^n = \frac{y^u}{h^u}$$

and

$$x^u = x^n \cdot w^u \qquad y^n = y^n \cdot h^u$$

Within the above calculations precision equivalent to a float shall be used. Conversion to integer shall only be used when the result is a point in a pixel oriented co-ordinate space (e.g. user space).

The resolution of an HGraphicsDevice is specified using the constant HScreenConfigTemplate.PIXEL_RESOLUTION with a Dimension on the setPreference(int,Object,int) method.

The constants HSceneTemplate.SCENE_PIXEL_LOCATION and HSceneTemplate.SCENE_PIXEL_DIMENSION allows applications to define the position and size of an HScene in the coordinate system of the HGraphicsDevice (in pixels). The constants HSceneTemplate.SCENE_SCREEN_LOCATION and HSceneTemplate.SCENE_SCREEN_DIMENSION allows applications to define the position and size which the HScene should occupy on the screen in normalized coordinates.

The combination of these defines the transformation between graphics pixels and the video output from the GEM device concerned. Only a limited set of transformations are required to be supported.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 248 of 861

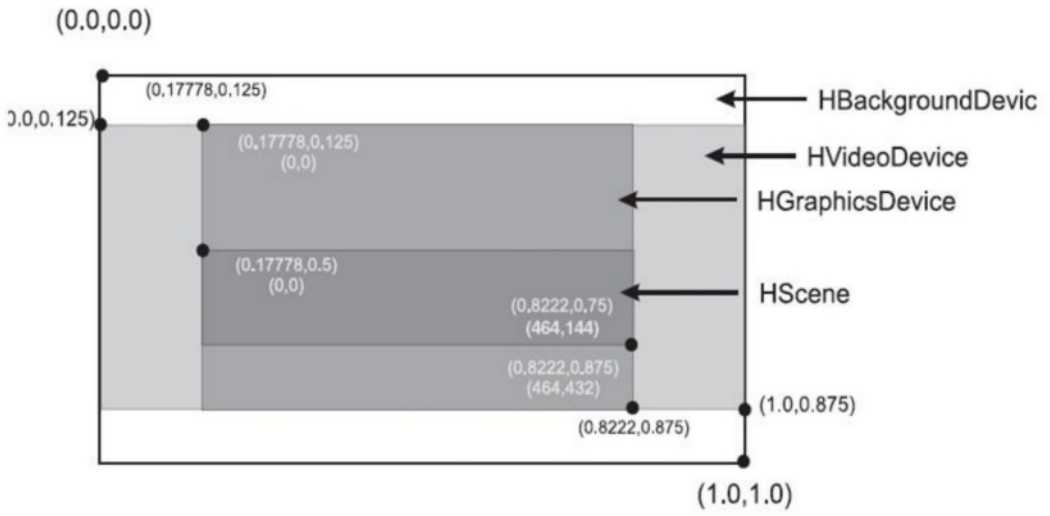


Figure 20: Possible configuration of HAVi Devices

Figure 20 shows a possible configuration of HAVi Devices. The `HBackgroundDevice` is configured to be full screen, the `HVideoDevice` to cover the area of (0.0, 0.125) to (1.0, 0.875).

When positioning video implementations may snap the video position to an adjacent line vertically (for example, to accommodate video and display field order) or an adjacent pixel horizontally (for example, to accommodate display chroma structure). The direction of "snapping" shall always be to minimize the error relative to the requested coordinate.

In figure 20 the `HGraphicsDevice` is configured to cover the area (0.17, 0.125) to (0.82, 0.875) with a pixel resolution of 464 x 432.

The `HScene` can be configured by setting the location of its origin and by defining its dimensions. The location of the `HScenes` origin can be specified by setting the preference `HSceneTemplate.SCENE_PIXEL_LOCATION` with a `Point` of (0,216) or setting the preference `HSceneTemplate.SCENE_SCREEN_LOCATION` with a `HScreenPoint` of (0.17, 0.5). The dimensions can be configured by setting the preference `HSceneTemplate.SCENE_PIXEL_DIMENSION` with a `Dimension` of (464, 144) or setting the preference `HSceneTemplate.SCENE_SCREEN_DIMENSION` with a `HScreenDimension` of (0.64, 0.25).

In some implementations and/or configurations pixels in the `HGraphicsDevice` may not correspond to discrete physical pixels in the actual display device. For example, this may be the case when the `HGraphicsDevice` is emulated.

13.2.1.3 Pixel Aspect Ratio

A pixel orientated coordinate system does not say anything about the pixel aspect ratio. The pixel aspect ratio (AR_x^{pixel} / AR_y^{pixel}) is defined by the aspect ratio of the display ($AR_x^{display}$ / $AR_y^{display}$, 4:3 or 16:9), the area that is covered by the `HGraphicsDevice` (w^n, h^n) and the pixel resolution of the `HGraphicsDevice` (w^u, h^u) (see figure 21).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

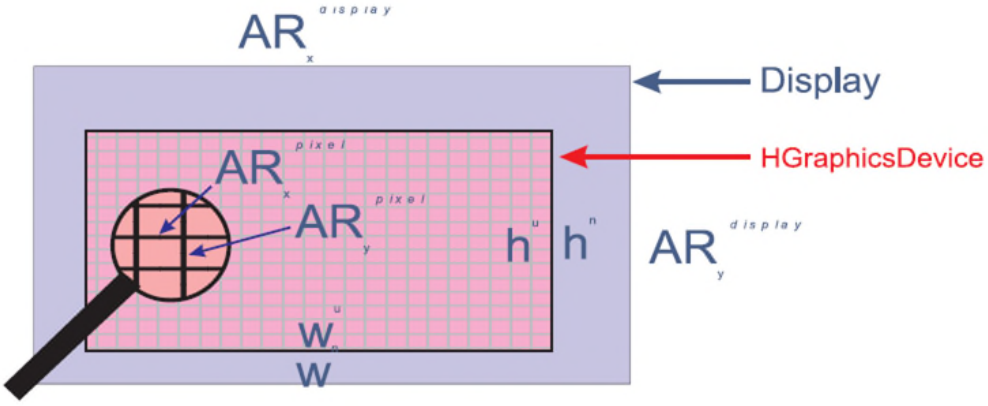


Figure 21: Calculating the Pixel Aspect Ratio

Table 58 shows typical resolutions of a full screen `HGraphicsDevice` and the corresponding pixel aspect ratios for different display aspect ratios. The supported resolutions are defined in annex G, "Minimum platform capabilities". Table 58 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.

Table 58: Typical Resolutions and their pixel aspect ratio (informative)

	4:3 Display	16:9 Display
Resolution for full screen HGraphicsDevice	**Pixel Aspect Ratio**	**Pixel Aspect Ratio**
640 x 480	1:1	4:3
720 x 480	8:9	32:27
720 x 576	48/45	64/45
768 x 576	1:1	48:36
854 x 480	3:4	1:1
1 024 x 576	36:48	1:1

13.2.1.4 Video space

The coordinate space for video is that defined by the input video signal after any scaling required by the platform (e.g. that required by ETR154). Scaling and clipping of video can be achieved using the JMF control `org.dvb.media.VideoPresentationControl`. The JMF control `VideoFormatControl` allows applications to query the various transformations being performed on video as part of its decoding and presentation.

13.2.2 Dependencies between HAVi screen device configurations (informative)

13.2.2.1 Principles

Under some circumstances there are dependencies between configurations of the various HAVi screen devices that are composed together to be output as a single screen. Two example reasons for this are the following:

- The video output signal for a screen must have a single aspect ratio - either 4:3 or 16:9 for standard definition GEM terminals. As a consequence, all HAVi screen devices in that screen must be configured to the same aspect ratio.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- Hardware limitations on memory bandwidth may limit the possibilities of combining HAVi screen device configurations. For example, some devices may be able to support the (optional) graphics resolution of 1 920 x 1 080 only under limited (device specific) conditions, e.g. not simultaneously with the decoding of MPEG-4/AVC video.

- Limitations on scalers may limit the possibilities of combining HAVi screen device configurations. For example, software based scaling solutions may not be able to simultaneously support full screen graphics resolutions of 960 x 540 and 720 x 576 in a single hardware graphics plane.

The concept of sets of HAVi screen device configurations which can be simultaneously selected is supported through the method HScreen.getCoherentScreenConfigurations.

13.2.2.2 Usage examples for standard definition, 625-line markets

If the aspect ratio of the video output signal is 4:3 then the configuration of each HAVi HScreenDevice that contributes to that signal must have a configuration shown in table 82, "Sets of coherent configurations for standard definition - 625-line markets" as having a screen aspect ratio of 4:3. Similarly if the aspect ratio of the video output signal is 16:9 then the configuration of each HAVi HScreenDevice that contributes to that signal must have a configuration shown in table 82, "Sets of coherent configurations for standard definition - 625-line markets" as having a screen aspect ratio of 16:9. DVB-J applications can obtain instances of such coherent configurations by creating HScreenConfigTemplate instances with PIXEL_ASPECT_RATIO set to either 16/15 (for 4:3) or 64/45 (for 16:9) depending on what is required.

13.2.2.3 High definition usage examples

The minimum requirements for support of HD are defined in clause G.1.1.3, "High definition" of the present document. These requirements result in 5 coherent sets of HAVi screen device configurations. These are shown below for 625-line/50 Hz markets. For 525-line/60 Hz markets, 720 x 576, 64/45 and 56/45 should be replaced by the values in table 83 "Sets of coherent configurations for standard definition - 525-line markets".

Set 1 (HD in front of SD, SD is 16:9):

- HGraphicsConfiguration[0] includes PIXEL_RESOLUTION = 1 280 x 720.

- HGraphicsConfiguration[1] includes PIXEL_RESOLUTION = 720 x 576 and PIXEL_ASPECT_RATIO 64/45.

Set 2 (HD in front of SD, SD is 14:9):

- HGraphicsConfiguration[0] includes PIXEL_RESOLUTION = 1 280 x 720.

- HGraphicsConfiguration[1] includes PIXEL_RESOLUTION = 720 x 576 and PIXEL_ASPECT_RATIO 56/45.

Set 3 (SD in front of HD, SD is 16:9):

- HGraphicsConfiguration[0] includes PIXEL_RESOLUTION = 720 x 576 and PIXEL_ASPECT_RATIO 64/45.

- HGraphicsConfiguration[1] includes PIXEL_RESOLUTION = 1 280 x 720.

Set 4 (SD in front of HD, SD is 14:9):

- HGraphicsConfiguration[0] includes PIXEL_RESOLUTION = 720 x 576 and PIXEL_ASPECT_RATIO 56/45.

- HGraphicsConfiguration[1] includes PIXEL_RESOLUTION = 1 280 x 720.

Set 5 (960 x 540):

- HGraphicsConfiguration[0] includes PIXEL_RESOLUTION = 960 x 540.

- HGraphicsConfiguration[1] is not defined in GEM.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 251 of 861

The following partial code fragment illustrates how the HAVi APIs can be used by DVB-J applications to ask for set #3 above and then configure the graphics system in that way.

```
HScreen s ;
HGraphicsConfigTemplate front = new HGraphicsConfigTemplate();
HGraphicsConfigTemplate rear = new HGraphicsConfigTemplate();
front.setPreference(
        HScreenConfigTemplate.PIXEL_RESOLUTION, new Dimension(720,576),
        HScreenConfigTemplate.REQUIRED);
front.setPreference(
        HScreenConfigTemplate.PIXEL_ASPECT_RATIO, new Dimension(64,45),
        HScreenConfigTemplate.REQUIRED);
rear.setPreference(
        HScreenConfigTemplate.PIXEL_RESOLUTION, new Dimension(1280,720),
        HScreenConfigTemplate.REQUIRED);
HGraphicsConfigTemplate input[] = { front, rear };
HScreenConfiguration c[] = s.getCoherentScreenConfigurations( input );
s.setCoherentScreenConfigurations(c);
```

For the case of set #5 above, HGraphicsConfiguration[1] is not defined. Some implementations may set the rear graphics plane to the DISABLED HGraphicsConfiguration. Other implementations may support 960 x 540 simultaneously with other graphics resolutions. Applications wishing to select set #5 should not constrain the rear graphics plane to DISABLED otherwise the call to getCoherentScreenConfigurations may fail on more capable GEM terminals.

13.2.2.4 Scalability and extensibility

GEM terminals may support additional HAVi screen devices and configurations beyond the minimum requirements defined in clause G.1.1, "Device capabilities" of the present document. No particular issues arise where the minimum required HAVi screen devices support additional configurations. GEM applications may discover those configurations and, if they are able to reserve the device concerned, attempt to set the configuration. The conditions under which setting such an additional configuration succeeds is not defined by GEM and may be very specific to a particular GEM terminal.

Where extra HAVi screen devices are supported in addition to the minimum requirements, these extra devices shall only appear behind the minimum required HAVi screen devices of the same type. GEM terminal implementations may support extra graphics or video devices in front of the minimum required HAVi screen devices which are not visible to the GEM environment, e.g. for transient UIs relating to device setup, configuration or operation.

13.3 Graphics

13.3.1 Modelling of the GEM display stack composition

The following clauses describes the theoretical model of the GEM display stack. Unfortunately, certain real world constraints may apply see clause 13.6, "Approximations". The "Graphics, Video and Subtitle pipeline" is illustrated in figure 22.

Figure 22 shows the conceptual model of the "graphical content" pipelines from an applications point of view. There are four sources of graphical content: subtitles, background, video and application graphics, and these four sources are mapped in a controllable way onto single screen. The figure shows the conceptual processing functions, the content stages, and application control points inside the content pipelines.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

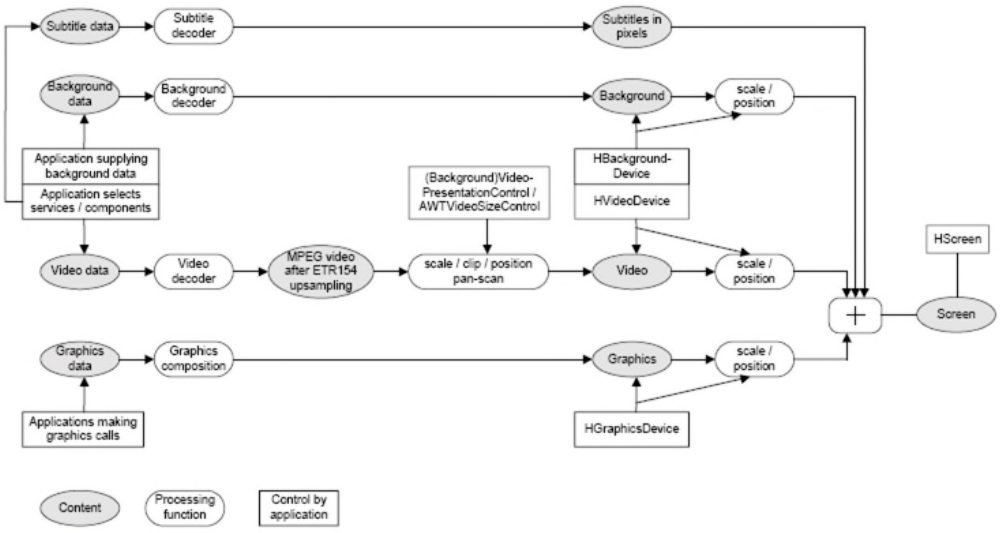


Figure 22: Graphics, Video and Subtitle pipeline

The figure shows that applications have full control over the source of the content that enters the different pipelines. Furthermore the application may control clipping, scaling and positioning of the content at different stages in the different pipelines. Of course an application can only apply operations that are supported by the underlying system.

The clipping and positioning of subtitles follows the video. Behaviour of subtitles when video is scaled is platform dependent.

From the application author's point of view the behaviour of the graphics composition process has 3 elements:

- The graphics elements are composed following the traditional graphics model using Porter-Duff rules (see Porter-Duff).The default rule used is the SRC_OVER rule.
- The background and video planes are composed using the Porter-Duff rule SRC_OVER.

NOTE: Only alpha of 0 and 1 is used in the Video planes.

- The results are composed together using the SRC_OVER rule (with the graphics results as the source and the results of the background / video composition as the destination).

This is illustrated in figure 23.

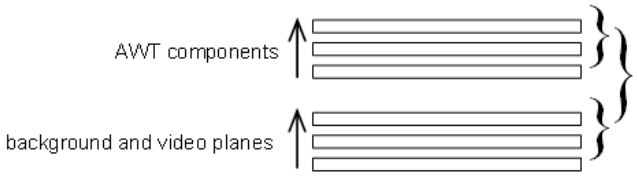


Figure 23: Overview of AWT / HAVi plane composition

The HAVi stack has a single full screen `HBackgroundDevice` at the back. In front of this are an ordered set of zero or more `HVideoDevices`. Each of the video devices can occupy the full screen area or part of it. Any area that is not occupied behaves as transparent. The occupied areas (video pixels) are considered to be opaque and will be obscured by any video devices in front of them. Non active video devices are invisible.

Systems that support only a single video device that can display full screen (and possibly other partial screen video devices) should report the full screen capable device as the first (back most) of the video devices.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The composition rules in each group follow the traditional "painter's" algorithm i.e. composing the layers from back to front.

The Xlet AWT Root Component is transparent (as shown in figure 24) so by default the result of the HAVi video and background device composition is the background to any application graphics.

Video already running in the HVideoDevice will keep on playing when an application starts.

The stack is illustrated in figure 24.

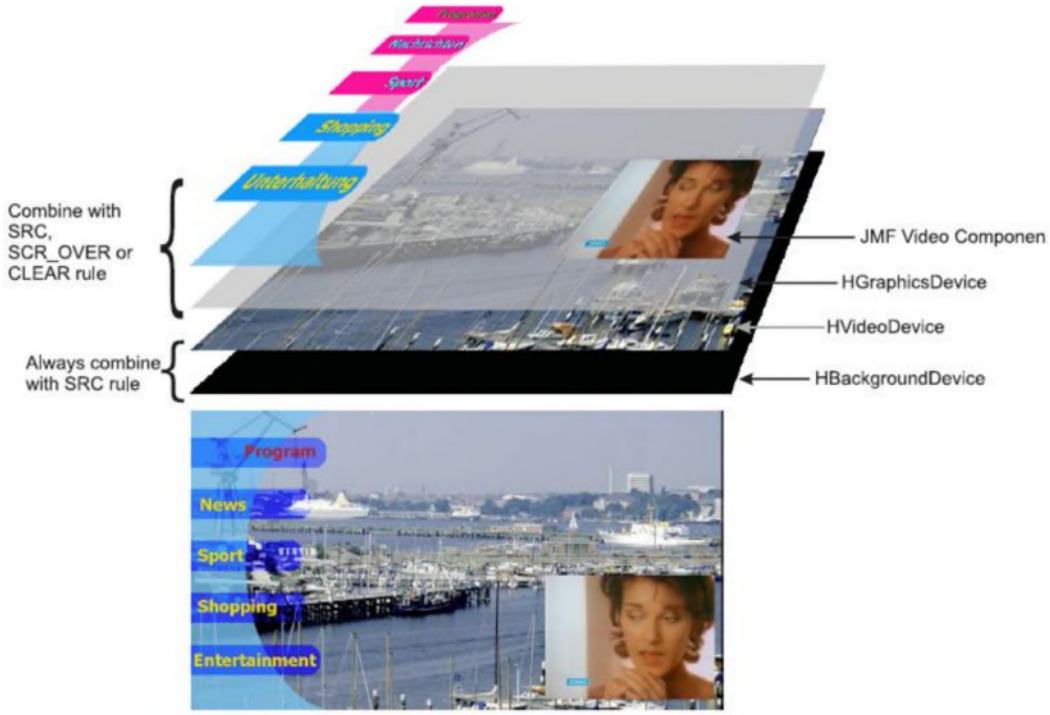


Figure 24: The GEM display stack illustrated

13.3.2 AWT Reference Model in the GEM

In the AWT all graphics rendering of the lightweight components is done via the `java.awt.Graphics` class. When a component needs to be redrawn it issues a repaint command which gets passed from component to component from top to bottom until a heavyweight component is reached. Although the enhanced and interactive profiles of GEM do not require any heavyweight components to be present, the `HScene` can be thought of being similar to a heavyweight component. Thus in GEM devices the repaint command gets passed until it reaches the root container - the `HScene`. The repaint method of the `HScene` causes a call to this component's `update` method as soon as possible. Before calling the paint method of a child the origin gets translated to the coordinate system of the child and the graphics object passed is clipped to the size of the child (see figure 25).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

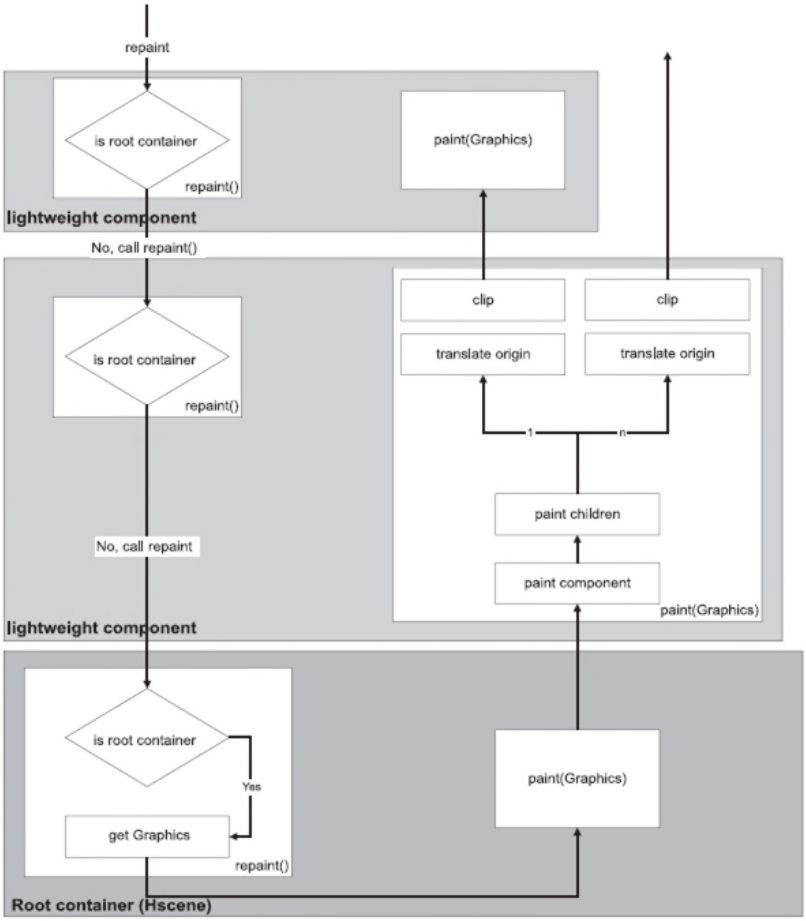


Figure 25: Repaint model in a GEM application

13.3.3 HAVi devices and AWT components

The top level user interface container for DVB-J applications is represented by the `org.havi.ui.HScene` class. DVB-J applications may obtain an instance of this class from an instance of `org.havi.ui.HSceneFactory`. The `HSceneFactory` class provides a number of methods which may be used to obtain an `HScene` depending on the specific requirements of the DVB-J application. `HScene` is conceptually equivalent to `java.awt.Window` or `java.awt.Frame` however it models the differing graphical environment found in many digital TV receivers. In this environment, there are typically significant constraints on what sizes and positions of top level containers may be possible.

One means for obtaining an `HScene` from an `HSceneFactory` is to populate an `HSceneTemplate` with a set of constraints and then request an `HScene` from the `getBestScene()` method which meets these constraints. Applications wishing to obtain a full screen sized `HScene` may use the `getFullScreenScene()` method specifying a particular graphics device on which the full screen scene should be presented.

GEM terminals are allowed to support only non overlapping `HScenes`. In implementations not supporting overlapping `HScenes` the following behaviour shall be implemented:

- Overlapping at creation or size change through `HSceneFactory`:
  - Returns a null reference if "`REQUIRED`" is used. Returns best effort if "`PREFERRED`" is used.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- Overlapping later when sizes change by awt:
  - Size does not change. Fails silently.

The GEM specification provides a general model for video output devices using the model found in the HAVi specification. A final output video signal is expressed through the `org.havi.ui.HScreen` class and is the result of adding a number of different video components.

- The output of one or more graphics decoders.
- The output of one or more video decoders.
- The output of any special decoders, e.g. a background.

An abstraction of each of these decoders is represented by an instance of a class inheriting from `HScreenDevice` - `HGraphicsDevice`, `HVideoDevice` and `HBackgroundDevice` respectively. Each of these can potentially exist in a range of configurations which are exposed by instances of classes inheriting from `HScreenConfiguration` - `HGraphicsConfiguration`, `HVideoConfiguration` and `HBackgroundConfiguration` respectively. Where devices support multiple configurations, applications may construct and populate templates to define criteria to select between configurations. These templates are classes inheriting from `HScreenConfigTemplate`.

As well as the general features, some of these sub-classes provide support for specific features of the devices concerned. `HGraphicsConfiguration` supports loading of images and listing of fonts specific to that configuration. It also support conversion between various coordinate systems. `HVideoDevice` provides access to the source of the video currently being decoded (a Locator) and provides access to the decoder object for the video currently being decoded (a `javax.media.Player`). The `HBackgroundClass` provides a means to support MPEG I-frames through the `HStillImageBackgroundConfiguration` class.

The method `HScreen.getCoherentScreenConfigurations ( HScreenConfigTemplate[] configs )` allows applications to express a common set of constraints for video, graphics and backgrounds and get back a coherent answer. In `HGraphicsConfigTemplate`, the constant `VIDEO_MIXING` allows applications to request configurations where graphics is super-imposed above video but without any requirement for pixels to be aligned. In `HScreenConfigTemplate`, there are constants to allow applications to ask for configurations as follows:

- `VIDEO_GRAPHICS_PIXEL_ALIGNED` - video and graphics pixels are the same size and aligned.
- `ZERO_VIDEO_IMPACT` - a graphics configuration must not change the existing video configuration.
- `ZERO_GRAPHICS_IMPACT` - a video configuration must not change the existing graphics configuration.

13.3.3.1 Video and graphics pixel aligned

The `VIDEO_GRAPHICS_PIXEL_ALIGNED` relationship between the pixels in the `HGraphicsDevice` and the pixels in the `HVideoDevice` is shown in figure 26.

NOTE: The relationship applies to the pixels in `HVideoDevice` after "Decoder Format Conversion" processing, and the pixels in `HGraphicsDevice`. As a result of this relationship the following constraint holds for the configurations of `HGraphicsDevice` and `HVideoDevice` that have aligned video and graphics pixels:

- The pixel aspect ratio of the pixels in both devices is equal.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

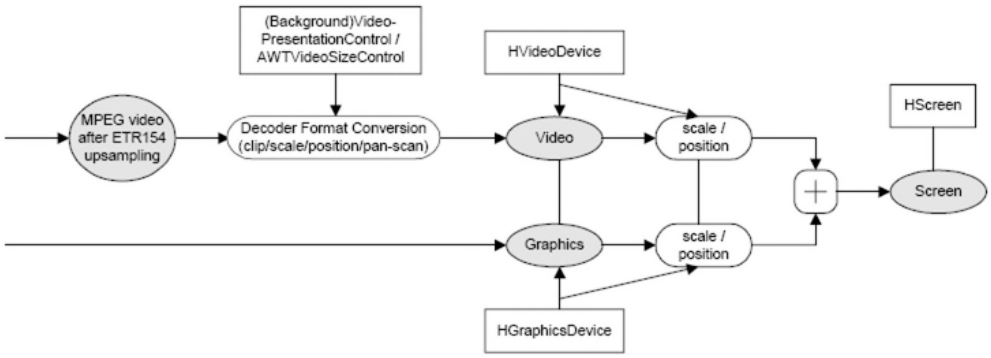


Figure 26: Video and graphics pixel impact

13.3.3.2 Zero graphics impact

Constants `ZERO_GRAPHICS_IMPACT` and `ZERO_VIDEO_IMPACT` can be used by applications to prevent changes to the `HGraphicsDevice` or the `HVideoDevice` respectively, in the case that changes are not intended. In general a change of the configuration of the `HGraphicsDevice` may lead to an automatic change of the configuration of, for example, the `HVideoDevice` (if the application is authorized to make this change), because restricted systems may not be able to deal with the two different configurations simultaneously. Therefore an application must specify `ZERO_GRAPHICS_IMPACT` or `ZERO_VIDEO_IMPACT` in the configuration template if it does not want to change the configuration of another device.

13.3.4 Composition

13.3.4.1 AWT paint rule

The normal AWT paint rules shall be followed. That is the root container (the `HScene`) is painted and then its components are painted recursively.

The observed *behaviour* shall be such that of each component drawing directly into the root component. Any use of temporary storage or double buffering shall not affect the final result of the AWT composition or its subsequent composition with the result of the video composition.

The process is shown in figure 27. The numbers in the arrows indicate the chronological order of the painting.

NOTE: The root container is painted first.

If a container has multiple components they get painted in the order N, N-1, ,0 where N indicates the order components are added to the parent container. (See Documentation on `java.awt.Container` in PBP 1.1 [4]).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

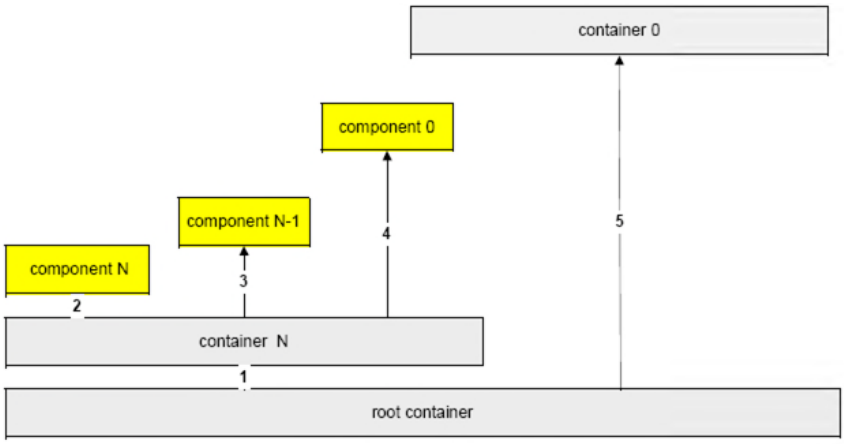


Figure 27: Chronological order of painting

13.3.5 Composition Rules

13.3.5.1 Components generally

By setting an appropriate `DVBAlphaComposite` rule on a `DVBGraphics` and using translucent colours blending may be achieved. By repeated placement of components complex scenes can be described. In figure 28 the background picture (of the harbour) is representative of the video plane. The translucent grey rectangle is drawn into a transparent off-screen buffer using the SRC rule. The red circle is then drawn into this off-screen buffer using one of the 8 most common Porter-Duff operations. When the AWT rendering in the off-screen buffer is complete it is composited with the video using the SRC_OVER rule.



Figure 28: Summary of the Porter-Duff rules

13.3.6 Extensions to the AWT graphics capabilities

For graphics the GEM specification mainly uses AWT facilities defined in PBP 1.1 [4]. However, certain extensions are made.

In order to allow compositing in GEM applications, the graphics capabilities of PBP 1.1 [4] have been extended by providing the `org.dvb.ui` package.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The package consist of the classes DVBGraphics, DVBAlphaComposite, DVBBufferedImage, DVBColor and one Exception. See annex U, "Extended graphics APIs".

13.3.6.1 Graphics Objects in GEM applications

The class org.dvb.ui.DVBGraphics extends the normal java.awt.Graphics class by adding support for alpha compositing.

In a GEM application each platform-created graphics object shall be an instance of org.dvb.ui.DVBGraphics.

DVBGraphics and DVBAlphaComposite has a principle support of 8 different Porter-Duff compositing rules but not all rules have to be supported for all DVBGraphics Objects.

Different DVBGraphics Object could support different compositing rules. E.g. a DVBGraphics Object created using a DVBBufferedImage with an image type of DVBBufferedImage.TYPE_ADVANCED supports all 8 specified Porter-Duff rules, TYPE_BASIC supports only SRC, CLEAR, SRC_OVER. Application can query the available compositing rules using DVBGraphics.getAvailableCompositingRules(). When setting an unsupported compositing rule a org.dvb.ui.UnsupportedDrawingOperationException will be thrown. All images whose getGraphics method returns a non-null object shall support alpha.

The supported compositing rules are defined in annex G, "Minimum platform capabilities".

The default compositing rule used by all graphics objects is SRC_OVER.

13.3.6.2 Buffered Image

The class DVBBufferedImage in the package org.dvb.ui adds the support for an accessible, transparent Image which can be used e.g. for off screen buffers. Two different platform dependent sample models are supported by DVBBufferedImage. When doing compositing in the TYPE_BASE sample model approximations may be applied (using SRC instead of SRC_OVER, see figure 36 or by approximating the alpha, see annex G, "Minimum platform capabilities") while in the TYPE_ADVANCED sample model the compositing rules set by the program will be used.

TYPE_ADVANCED is always a direct colour model while TYPE_BASE can be a CLUT based colour model.

13.3.6.3 DVBColor

NOTE: The general philosophy of this class has been to imitate the features of JDK 1.2 dealing with colour.

Unless explicitly specified otherwise, the internal implementation of the platform (e.g. java.awt) shall use the org.dvb.ui.DVBColor class instead of the java.awt.Color where technically possible (e.g. not the constructor for java.awt.Color). The class signatures shall not change. For example, where a method is specified to return java.awt.Color it shall return an instance of org.dvb.ui.DVBColor.

13.3.6.3.1 Modified packed colour representation

The most significant byte of the integer representation of an RGB colour is defined to hold an alpha value as is illustrated in figure 29. This data type is used in various of the constructors and methods described in the API documentation. It is referred to in the API documentation as TYPE_INT_ARGB.

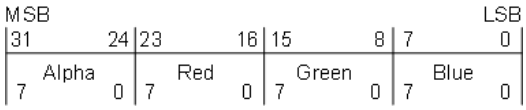


Figure 29: GEM colour format (TYPE_INT_ARGB)

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 259 of 861

13.3.7 14:9 Aspect Ratio Support

Support for the 14:9 aspect ratio is optional in GEM terminal specifications.

As well as the real display aspect ratios of 4:3 and 16:9, GEM defines a 14:9 aspect ratio. When this is in effect, all text will be subject to moderate aspect ratio distortion. However, broadcasters will be able to predict text flow without having to author specifically for each display type.

All `HGraphicsDevices` must be `HEmulatedGraphicsDevices` capable of emulating a14:9 aspect ratio on their actual aspect ratio at the time.

NOTE 1: Applications can still set the configuration of an `HEmulatedGraphicsDevice` to an `HGraphicsConfiguration` and have this be supported without emulation.

Where the current configuration of an `HEmulatedGraphicsDevice` has a 14:9 aspect ratio and the device is not reserved by any GEM application, the GEM terminal shall automatically maintain this 14:9 aspect ratio if the real underlying aspect ratio changes between 4:3 and 16:9 or vice-versa.

NOTE 2: The `HGraphicsConfiguration` observed by an application when it starts running may be unpredictable since there may be an already running GEM application which has reserved the `HGraphicsDevice` and set the `HGraphicsConfiguration`.

Setting the configuration of an `HGraphicsDevice` to an `HEmulatedGraphicsConfiguration` supporting a 14:9 display shall only impact the transformation of outline fonts into display pixels. Display pixels shall remain mapped 1:1 onto the actual pixels of the `HGraphicsDevice`.

13.3.8 Interaction between graphics from GEM and resident applications

If application graphics are wholly or partly obscured by graphics from resident applications (e.g. the GEM navigator), then either:

a) the resident application shall restrict itself to the use of opaque colours; or

b) the composition between the graphics of the resident application and those of GEM applications shall be done such that it does not result in video pixels that are obscured by opaque graphics pixels from a GEM application becoming visible.

Implementations that cannot meet one of these requirements shall remove the GEM application graphics from the screen when the graphics of resident applications are visible as described in clause 9.10, "Lifecycle interactions between GEM and resident applications".

13.4 Video

13.4.1 Component-based players and background players

Video is received by a GEM application as an MPEG sequence of compressed frames, each of which contains a large number of picture elements (pels) arranged in rows and columns. The GEM application decodes the MPEG stream to a presented stream which may be placed either in the video plane, or in a component in the graphics plane.

These two different ways of presenting video result in having two different kinds of JMF players, a background JMF player and a component-based JMF player.

A component-based JMF player plays video inside an AWT component, and the video inside that component is positioned and scaled by positioning and resizing the component. The video is always scaled to the full size of the component (or, if the terminal is not capable of scaling and positioning video this accurately, the video may extend outside the bounds of the component by up to one pixel in each direction [above, below, left and right]). This approximation shall not affect the position and size of the AWT component itself and hence shall not be visible to applications via the usual AWT methods).

Support for background players is mandatory for broadcast streaming formats (see clause 7.2, "Media streaming formats").

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Background JMF players play video in the video plane, outside and independent of any AWT component hierarchy.

When a component-based player is stopped, the last displayed frame shall continue to be displayed until the underlying video decoder resource is re-used. The behaviour when the underlying video decoder resource is re-used is implementation dependent.

When a background player is in the realized state, the video plane concerned shall become transparent to expose what is behind it such as background plane(s). See figure 16 "Illustration of the different types of display planes".

In GEM terminals exposing through the GEM-defined API the simultaneous decoding of more than one broadcast video stream, only a single background player shall be supported at one time. Realising a second JMF player while an existing background player is in use shall result in the second player replacing the first as the background player. Applications wishing the older player to remain visible must convert it to a component-based player and make that visible.

NOTE: GEM does not mandate a particular broadcast streaming format, e.g. Standard Definition 25Hz MPEG-2 Video is not required by GEM. However, as described in clause 7.2.2 a GEM terminal specification has to include some mechanism for delivering audio and video programming in standard definition. For these formats, background JMF players are created.

13.4.2 Modelling MPEG decoding and presentation pipeline

Figure 30 illustrates the underlying format conversion control process in a TS 101 154 [2] compliant SD digital receiver. In this model the video decoder produces "full-screen" video from the MPEG data (i.e. the decoder resolves any sub-sampling in the MPEG broadcast). Subsequent "Decoder Format Conversion" adapts this for the display device taking account of:

- Broadcast meta data (aspect ratio, pan/scan and active format description).
- Display knowledge (4:3 or 16:9, and resolution).
- User preferences (e.g. display wide screen in a letterbox).

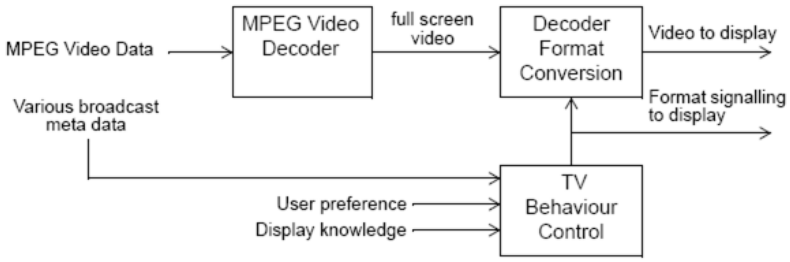


Figure 30: Format control in TV mode

If appropriate, the video that is sent to the display is accompanied by the proper WSS or SCART signalling to indicate the format of the video that is being sent.

NOTE: That the display device may do its own Decoder Format Conversion on top of the Decoder Format Conversion that the GEM device does. This is beyond the control of the GEM device.

In this version of the GEM specification, the video decoding process complies with clause 5.1.4 "Luminance resolution" of TS 101 154 [2] and applies up sampling for a defined set of luminance resolutions so that the decoded pictures are displayed at full-screen size. The input video source to JMF shall be the output of this up sampling process as illustrated in figure 31. In addition, there are two alternative sources of control for the "Decoder Format Conversion" process:

- Conventional TV format control behaviour (as in figure 30).
- Application format control behaviour.

The selection between these behaviours is under the control of the application. Before and after the existence of an application the behaviour is that for a conventional TV. During application execution the default behaviour of the JMF player's decoder format conversion shall be the conventional TV behaviour.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 261 of 861

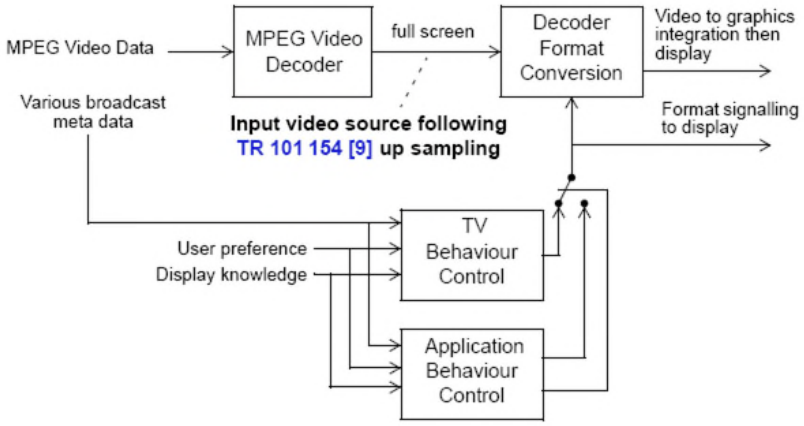


Figure 31: Format control in the presence of a JMF player

The Decoder Format Conversion consists of three steps that are performed on the full screen input video, which may have been up-sampled by the MPEG video decoder to become full screen. As illustrated in figure 32, these steps are clipping, scaling and positioning.

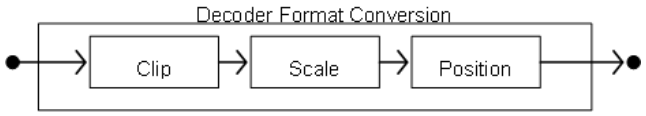


Figure 32: Reference model for Decoder Format Conversion

An application can query the implemented capabilities of each step of the Decoder Format Conversion using the org.dvb.media.VideoPresentationControl ("VideoPresentationControl"). For instance, it can query the supported scaling factors.

An application can also set up the Decoder Format Conversion steps atomically by using an org.dvb.media.VideoTransformation object ("VideoTransformation") that encapsulates the clipping, scaling, and positioning parameters. An application can get a number of pre-defined video transformations that correspond with standard Decoder Format Conversions like 16:9 letterboxing in a 4:3 display. It can either use these video transformations directly to set the Decoder Format Conversion, or it can change one or more parameters of the transformation before setting it. The API also offers support for querying the current video transformation.

13.4.3 Coordinate Spaces

The input to the Decoder Format Conversion block is always full screen video. If necessary, the MPEG video decoder block performs up-sampling as required by TS 101 154 [2] to get full screen video.

An application expresses the video clipping in terms of the pixel-based coordinate space of the full screen video. For 50 Hz SD video this is always a 720 x 576 raster, and for 60Hz SD video, 720x480 raster.

Positioning of the video is expressed in the normalized coordinate space for background JMF players. For component-based JMF players, the position of the video component is expressed in the pixel-based device coordinate space (i.e. a video component is positioned like any other AWT component).

13.4.4 Video components

A scaled portion of a video can be presented as a component within the AWT hierarchy. The controls that influence this presentation are parts of AWT and JMF.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Video components are treated just like any other component. However, it shall be noted that pixels within the video have opaque colours. The mapping to the source video is provided by the video presentation control (see annex N, "Streamed media API extensions") which allows an arbitrary portion of the video to be placed within the AWT hierarchy.

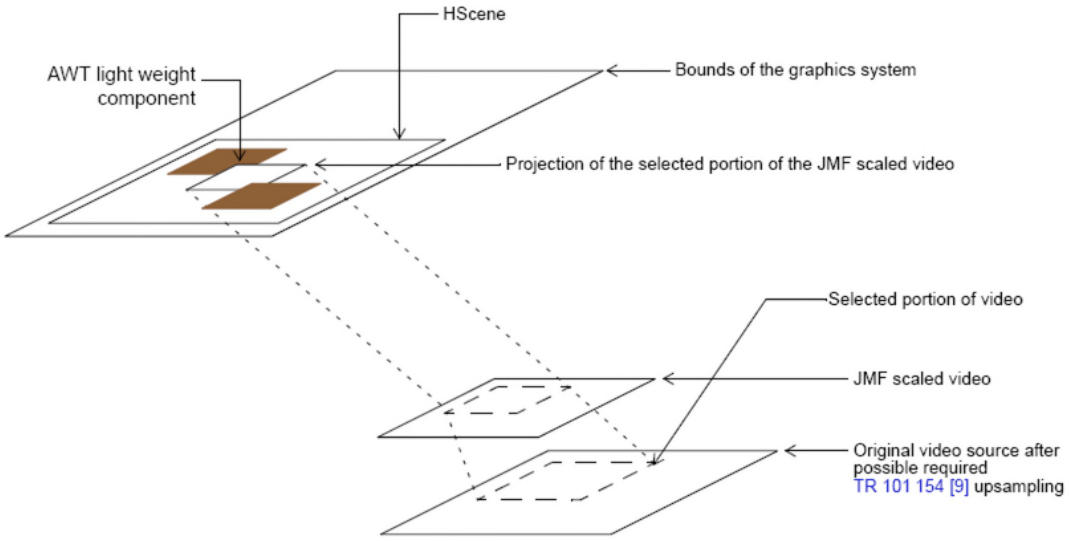


Figure 33: Introducing video into the AWT component stack

Graphics drawn over a component-based JMF Player using the SRC or CLEAR rules shall be alpha-blended with the scaled video from the component-based Player, not the background video plane.

Where two or more video components overlap, the z-order of the video shall be determined by the z-order of the video components within the AWT graphics system.

13.5 Subtitles

This clause only applies to GEM terminal specifications for which subtitling is available. If subtitling signalling is available, the presentation of subtitles shall follow the model specified here. Further, the introduction to subtitles contained in clause 13.1 and the components related to subtitles in figure 17 clause 13.2.1 apply only to GEM terminal specifications that include support for subtitling.

NOTE 1: US closed captioning is not the same thing as subtitles, thus, in systems where closed captioning is available but subtitling is not, this clause is optional.

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

13.5.1 Language and presentation setting

The following reference model shows the how the presentation of subtitles is controlled. The selection of the subtitles language depends by default on the user's preferences, the audio language and can be overridden by the application. The end user can set the subtitles on or off where the default depends on the preferences. The application can override all this and switch the subtitles off even if the user has set them on.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Figure 34 illustrates the decision procedure.

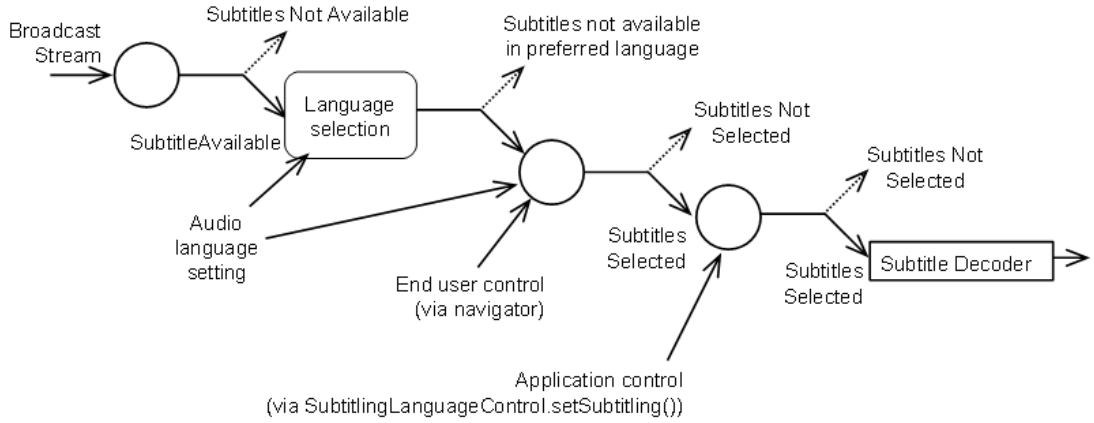


Figure 34: Determining subtitling language and presentation setting

The DVB-J API includes a control (see "SubtitlingEventControl") that the application can use to get notified of the state changes in the availability and presentation status of the subtitles. This corresponds to the status of the left hand side input and the right hand side output in the diagram. The language selection and the resulting preferred language are determined using an implementation dependent algorithm. e.g. the platform may support separate language preferences for audio and for subtitles.

The org.davic.media.SubtitlingLanguageControl allows the application to query the currently set subtitling language, override the default language setting and switch the subtitles off or let them be end user controlled. This corresponds to the language selection phase and the application controlled switch in figure 34.

13.5.2 Relation to graphics

Ideally, subtitles should be presented on top of the video plane but below the graphics plane(s). However, GEM terminals conforming to the present document are only required to support subtitles in areas where they are not overlapped by application graphics (i.e. by an HScene for a DVB-J application). If any GEM application has an HScene occupying the full area of the screen or has the AWT focus, GEM application graphics should be given preference over subtitles when they overlap otherwise subtitles should be given preference.

See also the description of television viewing mode in clause 11.4.1.4, "Television viewing mode".

The subtitling plane is full screen and allows the subtitles to be positioned anywhere on screen. The positioning of the subtitling texts is part of the subtitling stream content and is fully controlled by the broadcaster.

13.5.3 Coordinate Spaces

Subtitles are decoded into a plane with the same coordinate system and position as the `HVideoDevice`.

13.6 Approximations

13.6.1 Approximations in composition

The GEM specification references the Porter-Duff rules for composition (see Porter-Duff).The minimum set of operations required is defined in (clause G.1.3.1, "Composition rules"), in addition the implementation of the compositions may be approximated as detailed below.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

13.6.1.1 Implementation of modes

Typical implementations have a hardware blending process that blends the graphics plane over the video using SRC_OVER (see figure 35). This places certain practical limitations on the purity of the display model.

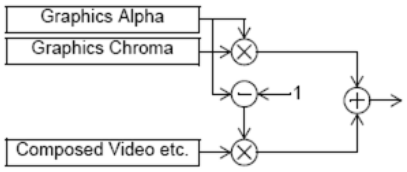


Figure 35: Typical current technology implementation

The SRC rule simply requires that the colour and alpha characteristics of the new pixels replace those previously present.

Components obtained from a JMF Player via the `Player.getVisualComponent()` method are treated as having an alpha of 1 so whether SRC or SRC_OVER is used when placing the video component the effect is that the video completely replaces anything previously drawn.

For the purposes of this discussion, MPEG I-frames and video drips shall be considered as video and not as graphics even if particular implementations may implement them using part of the graphics system of their hardware.

13.6.1.1.1 Graphics directly over video

When drawing graphics directly over video:

- The effect of SRC_OVER mode is as expected as the alpha value of the drawn graphics is used to control blending of the graphics with the video.

13.6.1.1.2 Graphics over other graphics

13.6.1.1.2.1 SRC

If graphics are painted over other graphics using SRC mode the result is as expected.

13.6.1.1.2.2 SRC_OVER

If graphics are painted over other graphics using SRC_OVER mode the result will be implementation dependent when $0 < \text{alpha} < 1$. Drawing with colour where alpha = 0 (i.e. fully transparent) shall not cause any effect to the existing pixels. Figure 36 illustrates a variety of implementations of SRC_OVER graphics to graphics blending. One allowed behaviour where the sample model is of `TYPE_BASE` is that defined in PBP 1.1 [4] for those signatures of the `drawImage` methods in `java.awt.Graphics` which do not have a `"bgcolor"` parameter. Specifically, only pixels which are fully opaque are drawn at all. Pixels which are transparent to any extent do not affect whatever pixels are already there.

a) Shows the logically correct result where the green and red areas mix to produce intermediate colours.

This implies a graphics to graphics blending process, it also implies a large gamut in both the chroma and alpha channels. This may not be practical in many early implementations.

The following cases illustrate simplifications of the blending scheme. These should not be considered equally good approximations:

b) Here the graphics alpha is preserved only when it is drawn directly over a video component. When drawn over another graphic the alpha facets of the colours are considered to be 1.

c) Here the source graphics alpha is preserved when it is drawn over a video component even if there is an intermediate semi-transparent graphic. Where the source is over opaque graphics then a graphic to graphic blend is implemented.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 265 of 861

d) Here the source graphics alpha becomes the hardware mixing alpha regardless of what has been drawn previously over the MPEG image. This approximation is not permitted for mixing of graphics from the same GEM application. This approximation is permitted for mixing of graphics between GEM applications on GEM terminals that support overlapping HScenes.

e) Here the source graphics alpha becomes the hardware mixing alpha in areas where the alpha is already less than 1. Where the alpha is currently 1 (i.e. over opaque graphics) the alpha remains 1.

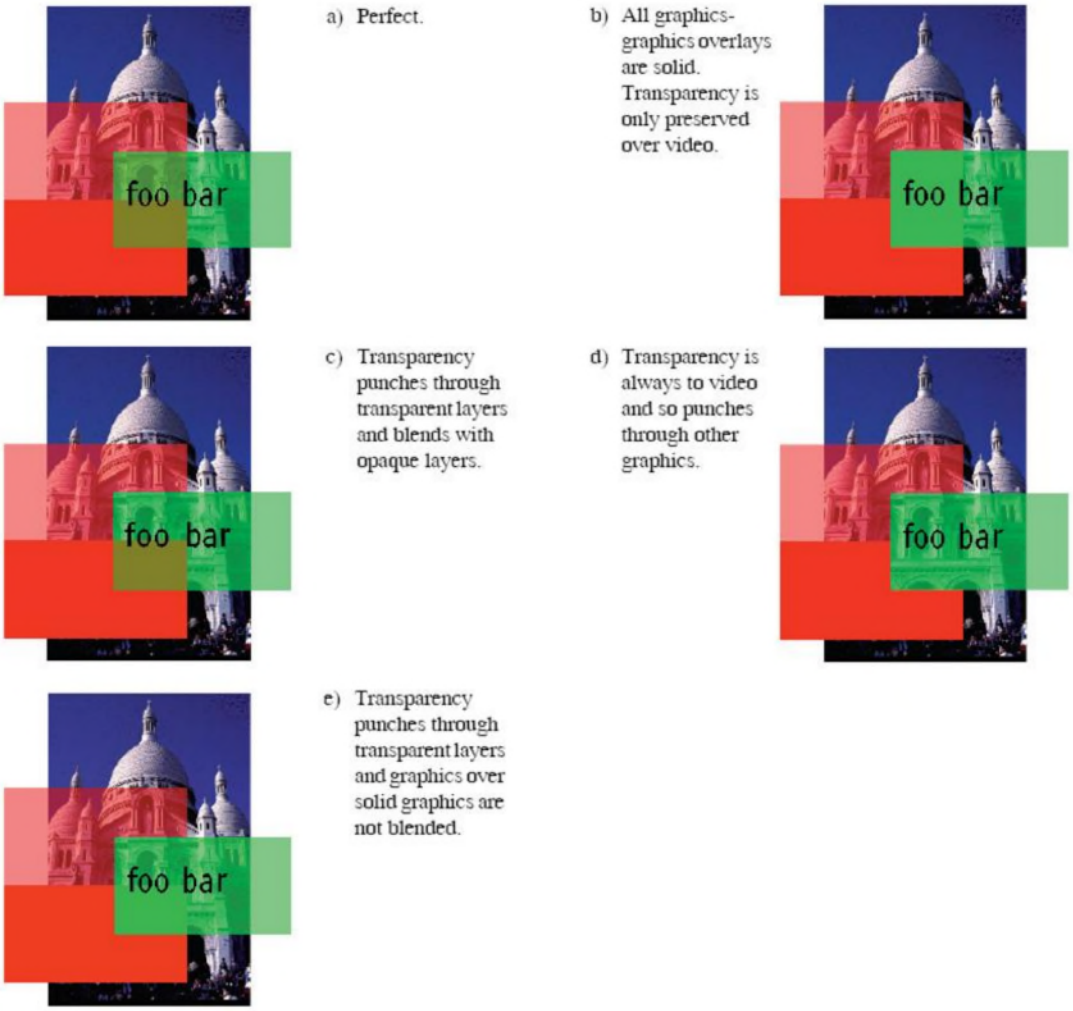


Figure 36: Implementations of blending

13.6.1.1.2.3 CLEAR

If graphics are painted over other graphics using the CLEAR mode the result is as expected. This operation is the same as using a source with alpha=0 and the SRC rule.

13.6.1.2 Approximation of alpha

The precision of implementation of alpha depends on the `DVBGraphics` object concerned. The minimum requirements are specified in annex G, "Minimum platform capabilities". The actual colour used for a given colour can be queried using `org.dvb.ui.DVBGraphics.getBestColorMatch()`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

13.6.1.3 Approximation of colour

Logically the colour model is a "true colour" one. However, colour approximation is allowed as described in clause G.1.5, "Colour capabilities".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 267 of 861

14 System integration aspects

14.1 Namespace mapping

GEM does not mandate any particular format for locators. Note, however, that clause 14.8, "Locators and content referencing" requires that terminal specifications based on GEM define some text representation for certain entities.

14.2 Reserved names

File names starting with the characters "dvb." are reserved for use as "well known" files defined in this or future specifications.

Authors shall not use file names with this form to avoid possible collision with standards defined files.

14.3 XML notation

These XML notation rules only apply to XML file formats defined in GEM.

These rules shall apply to the processing and encoding of all the files where XML is used as an encoding format in a GEM application.

Rules for encoding of the XML formatted files:

- The file shall be a well-formed XML document (but not necessarily valid against the DTD specified in this version of GEM). Here 'well-formed' and 'valid' are used as defined in the XML 1.0 (see XML 1.0 [60]) specification.

NOTE 1: The remark on validity is included, because it is possible to be valid only relative to one DTD. Valid documents relative to a DTD specified in a later version of GEM would not be valid relative to the DTD specified in the present document - however, the rules defined here intend to provide this future-proofness and allow terminal implementations compliant with the present document to be able to process files that may be encoded according to a later version of GEM.

- The XML files may contain the XMLDecl item ("<?xml ?>" tag) in the prologue in the beginning of the file.
- All the XML files shall be formatted using the UTF-8 character encoding which is the default used in XML.
- GEM terminal specifications may extend the allowed XML notation by permitting other character encodings, if this extension is explicitly stated in a GEM terminal specification. If no encoding is specified in an XML file, however, the default shall be UTF-8.
- The possible XMLDecl item in the beginning of the file shall not contain an 'encoding' attribute specifying another encoding than UTF-8.
- If the XMLDecl item is included in files conforming to the present document, it shall indicate XML version 1.0.
- The XML file shall contain a document type declaration ("<!DOCTYPE >" tag) where the Name is the same as the name of the root element.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- The document type declaration shall contain an `ExternalID` item with the "PUBLIC" identifier and both a `PublicLiteral` and a `SystemLiteral`. GEM specifies the `PublicLiteral` that shall be used to identify the document types defined in GEM. GEM specifies a `SystemLiteral` that can be used for identifying a location where the DTD can be retrieved. The `SystemLiteral` included in the document type declaration shall point to a location where the DTD can be obtained via the Internet using the HTTP protocol as specified in GEM.

- The `PublicLiteral` is used for identifying the type of the file. For document types specified in GEM, the `PublicLiteral` shall have the following syntax:

```
"-//DVB//DTD " <document type> " " <version_number> "//EN"
```

where:

```
<document type> has the following syntax:

    <document_type> = letter letters
    letters = "" | letter letters
    letter = uppercase_letter | lowercase_letter | space
    uppercase_letter =          "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H"
                                | "I" | "J" | "K" | "L" | "M" | "N" | "O"
                                | "P" | "Q" | "R" | "S" | "T" | "U" | "V"
                                | "W" | "X" | "Y" | "Z"
    lowercase_letter =          "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h"
                                | "i" | "j" | "k" | "l" | "m" | "n" | "o"
                                | "p" | "q" | "r" | "s" | "t" | "u" | "v"
                                | "w" | "x" | "y" | "z"
    space = " "

<version_number> has the following syntax:
    <version_number> = major_version "." minor_version
    major_version = digit digits
    minor_version = digit digits
    digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
    digits = "" | digit digits
```

The `<document_type>` part of the `PublicLiteral` is used as an identifier of the document type. When future versions of GEM specify newer, backwards compatible versions of the document type, the `<document_type>` part shall not be changed and the `<version_number>` part shall be changed to a new version number unused in a previous version of GEM for that document type.

- The XML file shall be valid relative to the DTD identified in the `ExternalId` document type declaration. Here 'valid' is used as defined in the XML 1.0 [60] specification.

- The document type declaration shall not contain a declaration part in square brackets ('[' and ']').

- Where the XML type PCDATA is used in XML elements and it is specified that this contains a string (for example a file name), these strings shall not contain '<', '&' or '>' characters that might be mistaken as XML tags or references of the markup.

NOTE 2: Strings where these characters are allowed should be specified to be encoded as CDATA, leading to a more complex notation but allowing those symbols to be used.

- The file shall include only tags and attributes that are defined in GEM or a later version of GEM.

- The file shall not include XML entity declarations ("<!ENTITY .>" tags).

- The file shall not include XML character or entity references (references starting with '&' character).

- The file shall not include XML processing instructions, except optionally the "<?xml ?>" XMLDecl item.

- The file may include XML comments ("<!-- -->" strings), but not within elements that are specified as PCDATA containing strings to be encoded as defined in GEM.

Rules for processing of the XML formatted files in the GEM terminal:

- The parser shall use the `PublicLiteral` in the document type declaration in the XML file ("<!DOCTYPE…>" tag) for identifying the type of the file.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 269 of 861

- The `PublicLiteral` in the document type declaration identifies the version of the DTD that is used for this file. There is no requirement for the parser to try to fetch that DTD file using the URL defined by the `SystemLiteral`. It is an implementation option for the parser to retrieve the DTD from that URL. If the DTD is unavailable from that URL, then the behaviour shall be platform dependent.

- The parser shall accept files that have a different version number in the `PublicLiteral` than the one specified in GEM for the given file type. These are probably files encoded according to a different version of GEM. From those files, the parser shall parse, recognize and handle all those elements and attributes that are part of the DTD included in the present document.

- The parser shall ignore all XML elements (start tag, end tag, and possible string between them) that are not specified in the DTD included in the present document.

NOTE 3: This allows extending the DTDs in the future in a future proof manner where existing terminals ignore all the elements introduced in later versions of GEM

- The parser shall ignore such attributes of XML tags that are not specified in the DTD included in the present document.

- The parser shall ignore XML comments encoded as defined in the XML 1.0 specification.

- The parser must accept empty XML elements specified in GEM both in their start-tag and end-tag form as well as in the empty element tag form (e.g. '<tuning value="true"></tuning>' may be used as well as '<tuning value="true"/>').

- Rules for evolving the specification must ensure that the encoding of the file will always be maintained backwards compatible when these rules are followed (i.e. later versions of GEM may add new XML tags and new attributes to existing XML tags, but may not change the semantics of the existing elements).

- If the encoding of the file violates the rules defined for the encoding above, the behaviour of the parser can be platform dependent, including the possibility that the parser may completely discard such files and the system may behave as if the file is not present at all.

14.4 Network signalling (error behaviour)

The behaviour of GEM terminals when receiving incorrectly formatted data, however transmitted or otherwise acquired, is implementation dependent except where a specific error behaviour is required by GEM or referenced specifications. GEM terminals may implement whatever strategy they like for this situation. It is an allowed implementation choice to pass values from the network straight through to applications without checking them for correctness. Hence API calls which are specified as returning a specific piece of information may not return a valid piece of information if the original information in the network is wrong.

GEM terminals should observe the behaviour defined in section 4.1 of TS 101 154 [2].

NOTE: It is highly recommended that the GEM terminal should be designed to allow for future compatible extensions to the DVB SI, DSMCC or other formatted data interpreted by the GEM terminal. All of the fields "reserved" (for ISO), "reserved_future_use" (for ETSI), and "user defined" in the EN 300 468 [14] should be ignored by GEM terminals not designed to make use of them. The "reserved" and "reserved_future_use" fields may be specified in the future by the respective bodies, whereas the "user defined" fields will not be standardized. Where a GEM API provides access to this data, the data should be returned to the GEM application without validation or correction.

14.5 Text encoding of application identifiers

Where an `organisation_id` or `application_id` is encoded in textual form it shall be encoded as follows:

- A hexadecimal representation of the value.

- Lower case letters.

- No extra leading zeros (as would be produced by Integer.toHexString).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Where both an organisation_id and application_id are combined into an application identifier, they will be represented as a single hexadecimal number using the previously described encoding with the organisation_id as the most significant bits and the application_id as the least significant bits.

14.6 Filename requirements

14.6.1 Persistent storage

Receivers shall support path and file names as specified by persistentpath and persistentfilename in the following BNF:

```
lowalpha =      "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" |
                "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" |
                "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"

upalpha =       "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" |
                "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" |
                "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"

alpha   =       lowalpha | upalpha

digit   =       "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

punct   =       "_"

persistentfilechar := alpha | digit | punct

persistentfilesubstring := persistentfilechar | persistentfilesubstring persistentfilechar

persistentfilename = persistentfilesubstring
                           | persistentfilesubstring "."
                           | persistentfilesubstring "." persistentfilesuffix
persistentpath = persistentfilesubstring | persistentpath "/" persistentfilesubstring

persistentfilesuffix = persistentfilechar | persistentfilesuffix persistentfilechar
```

Receivers are required to support:

- persistentfilesubstrings of length less than or equal to 8 characters.
- persistentfilesuffixes of length less than or equal to 3 characters.

Receivers are not required to reject filenames which exceed these requirements but applications using such filenames are not compliant. These restrictions do not apply to stored applications.

Receivers shall have filesystems for persistent storage which are either case sensitive or "case preserving". Applications shall be written to work on both of these. "case preserving" filesystems are case insensitive when opening an existing file but preserve the case which was used when the file was initially created.

14.6.2 DSMCC object carousel

Receivers shall support object carousels at least matching these requirements;

- File paths (i.e. the concatenation of directory names, directory separators and a filename) must be ≤ 1 024 bytes long.
- Within the file path requirement, file and directory names must be ≤ 255 bytes.
- The OC file system shall be case sensitive.

NOTE 1: There is no limit on the maximum number of directory nesting levels as long as the limit on file paths is not exceeded.

NOTE 2: Authors should avoid relying on OC being case sensitive since some authoring platforms will get it wrong.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

NOTE 3: To ease development of applications on computer platforms using traditional file systems the author is recommended to restrict file names to the character codes 0x21 to 0x7E excluding 0x22, 0x27, 0x3A, 0x3B, 0x5C (double quote, single quote, colon, semicolon, backslash).

14.6.3 Stored applications

Receivers shall support file and path names in stored applications as defined for the object carousel in clause 14.6.2, "DSMCC object carousel" except as follows:

- File systems for stored applications shall be either case sensitive or "case preserving". Applications to be stored shall be written to work on both of these. "Case preserving" is defined in clause 14.6.1, "Persistent storage" above.
- The characters 0x22 (double quote), 0x27 (single quote), 0x3A (colon), 0x3B (semicolon), 0x5C (backslash), 0x2F (slash), 0x3F (question-mark), 0x3C (less-than), 0x3E (greater-than), 0x2A (star) or 0x7C (pipe) shall not be used.

14.7 Files and file names

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

The GEM specification defines how applications can use the names of files in order to access content held in files. It is intentionally silent about the file systems and file system namespaces of GEM terminals except as defined below.

- When a GEM application starts, the filesystem where that application is carried will be mounted into the file system namespace of the GEM terminal concerned. For a DVB-J application clause11.5.1, "Broadcast Transport Protocol Access API" defines that creating a new instance of `java.io.File(".")` will result in a reference to the base directory of the application. The present document does not define the use of "file:" URLs relative to this base directory, only absolute URLs are defined. This base directory may be a sub-directory within this filesystem.
- GEM applications which have requested the right to access persistent storage, and had this right granted, are allowed to access the persistent file namespace. For DVB-J applications, the top level directory of this namespace is obtainable from the system property, `"dvb.persistent.root"`.
- GEM applications may have the ability to mount additional filesystems into the file system namespace of the GEM terminal concerned. DVB-J applications are allowed to use the `attach()` method on the `org.dvb.dsmcc.ServiceDomain` class in order to attach an object carousel as an additional file system. In all other methods, using a locator that refers to a file or directory, as described in clause 14.8, "Locators and content referencing" shall not mount the specified object carousel file system.
- Conformant GEM applications shall not attempt to access files or file systems outside what is allowed by the present document. The consequences should they attempt to do this are undefined and implementation dependent. Platforms are allowed to choose to limit the access rights of DVB-J applications through use of platform security mechanisms, e.g. `java.io.FilePermission`.
- References to content carried in files shall either be done using names of files encoded in text or using "file:" URLs as defined in RFC 1738 [61]. File names encoded in "dvb:" locators shall be transformed to "file:" URLs before use. For DVB-J applications, file names shall be encoded in Java String objects and "file:" URLs shall be encoded in instances of `java.net.URL`. See `ServiceDomain.getURL(Locator)`. When applications construct a `java.net.URL` for the "file:" protocol, the host part shall be empty and the path part shall be as specified for the constructors of `java.io.File` with the platform file separator replaced by "/" and omitting the first separator character.
- Stand-alone interaction channel applications shall see a file system where the current directory is derived from the contents of the AIT File (see clause 10.4.5, "AIT File"). Unless the GEM terminal is certain of the complete contents of a directory in this file system, all attempts to list such contents shall fail as if an I/O error had occurred in the underlying system.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 272 of 861

14.8 Locators and content referencing

Table 59 lists the types of entity that may be addressed by locators in GEM, and defines the entities for which a text representation is required. In the case of locators, where a text representation is required GEM does not specify what that representation is, however GEM terminal specifications shall supply an unambiguous, concrete syntax for each of these entities.

NOTE 1: Clause 11.7.6 "Content referencing" describes how this text representation can be used in DVB-J applications.

GEM does not require support for addressing any other type of entity in a GEM system by locator or URL.

Table 59: Addressable entities, locators and their text representation

Entity	Text Representation
Required for all targets:	
Service	locator text representation shall be defined
Service domain	locator text representation shall be defined
File	"file:" URL as defined in RFC 1738 [61] (note 1) DVB locator including "dvb_abs_path" element (note 2) "http:" URL as defined in RFC 2616 [45]. "https:" URL as defined in RFC 2818 [73] locator text representation shall be defined for files located within a Service domain
Directory	"file:" URL as defined in RFC 1738 [61] (note 1) DVB locator including "dvb_abs_path" element (note 2) "http:" URL as defined in RFC 2616 [45]. "https:" URL as defined in RFC 2818 [73].locator text representation shall be defined for directories located within a Service domain
Required for broadcast targets only:	
Transport stream	locator text representation shall be defined
Network	no standardized text representation required
Program Event	locator text representation shall be defined
Drip feed decoder	"dripfeed://"
Required for targets except IPTV:	
MPEG Elementary Stream	locator text representation shall be defined

NOTE 2: Even if an optional carousel functional equivalent is not defined, there still is a locator for files carrying the application.

14.9 Content referencing for IPTV/ OTT

14.9.1 Introduction (informative)

Java TV defines the concepts of transport independent locators and transport dependent locators. Services are equal if their Locators are equal which requires their external forms are equal. GEM permits both of the following;

- transport independent references to a service where there is no requirement for GEM applications to specify whether the service is available via a classical DVB network interface or an IP channel;
- transport dependent references to a service where GEM applications specify the network interface or channel by which the service is to be delivered.

In consequence, implementations may need to support at least 3 different external forms for locators - transport independent, transport dependent via classical DVB network interface and transport dependent via IP channel.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The SD&S broadcast discovery information record syntax permits the transport independent locator to be the basic "dvb:" locator as defined in MHP [1], clause 14.1, "Namespace mapping (DVB Locator)" since it contains enough information to resolve this locator. Hence this permits GEM applications using those locators to be independent of the means of their distribution. Transport dependent locators for content delivered via a classical DVB network interface need to be those defined in earlier versions of GEM for compatibility reasons.

Transport dependent locators for content delivered via IP need to identify the source of the transport stream and the service within that transport stream (for the case of MPTS). To identify the service within the transport stream, the DVB-SI service Id is sufficient since this is the same as the program number in the corresponding program map table.

14.9.2 Details

Table 60 shows the different forms of locators which are applicable to GEM.

Table 60: Locators and URLs

Type of reference	Class	External Form
Transport independent	org.davic.net.dvb.DvbLocator	"dvb:" as in MHP [1], clause 14.1, "Namespace mapping (DVB Locator)"
Transport dependent		
Service accessed via classic DVB network interface (see note 1)	org.davic.net.dvb.DvbNetworkBoundLocator	No standardized text representation
Service accessed via RTSP implemented in the client	org.dvb.locator.ip.RTSPLocator	"rtsp:" URL defined in clause 3.2 of RFC 2326 [83]
Service accessed via unicast IP (including RTSP implemented in a GEM application)	org.dvb.locator.ip.UnicastLocator	No standardized representation
Service accessed via multicast IP	org.dvb.locator.ip.MulticastLocator	No standardized representation (see note 2)
Service accessed via HTTP	org.dvb.locator.ott.HTTPLocator	"http:" URL defined in clause 3.2.2 of RFC 2616 [45]
Service accessed via HTTPS	org.dvb.locator.ott.HTTPLocator	"https:" URL as defined in clause 2.4. of RFC 2818 [73].
NOTE 1: Not applicable in GEM terminals without a classic DVB network interface. NOTE 2: The "rtp:" URI defined in clause 8 of TS 102 539 [81] may be used to obtain a MulticastLocator however this does not imply or restrict the content being accessed to being encapsulated in rtp. There is no requirement for MulticastLocator.toExternalForm to return a "rtp" URI.		

Where references to IP content use the "dvb:" URL defined in MHP [1], clause 14.1, "Namespace mapping (DVB Locator)" of the present document, these shall be matched against the DVBTriplet or TextualIdentifier fields in the broadcast discovery records defined by clause 5.2.6.2 of TS 102 034 [80].

A DvbNetworkBoundLocator is the transport dependent locator for services delivered by a classical DVB network interface. Even in GEM terminals without a classical DVB network interface, these shall not be matched against services announced by TS 102 034 [80].

14.10 Service identification

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

A transport independent locator has additional properties:

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

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

- They can give alternate identifications for a single service.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Terminal specifications based on GEM shall define a textual representation for at least one of the transport dependent or transport independent locators. A standardised format definition of the textual representation is beyond the scope of the present document.

14.11 CA system

GEM places no requirements on a CA system, if one is present.

NOTE: GEM terminal specifications may include the MHP requirements in MHP [1], clause 14.11 as described in clause 15.6, "Functional equivalents".

14.12 Focus management

NOTE 1: The following is basically aligned with the focus handling rules defined in clause 25.2.2.1.1 of OCAP [5].

The GEM terminal shall maintain an ordered list of those HScenes which are able to receive focus at the current time. HScenes shall be included in this list if and only if they meet all of the following criteria;

- they are visible (as defined by HScene.isVisible);
- they are active (as defined in HScene.setActive);
- they have explicitly requested focus at least one time;
- the HScene's application is in the Active state.

When an HScene not in the list meets all the criteria to be included in the list, it shall automatically be added to the end of the list. When an HScene in the list stops meeting all of these criteria, it shall automatically be removed from the list. Calling dispose() on an HScene in the list shall result in it being removed from the list.

When an HScene gets focus, it shall move to the start of the list pushing other HScenes down the list. The HScene with focus shall remain at the start of the list while it has focus. If the HScene with focus is removed from the list then the focus shall be given to the next HScene in the list.

NOTE 2: The class description of org.havi.ui.HScene defines requirements to report gaining and losing focus by applications using WindowEvents of type WINDOW_ACTIVATED and WINDOW_DEACTIVATED.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

15 Detailed platform profile definitions

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

All GEM platforms must support the following underlying platform specifications:

- CDC 1.1 [3] or later: the Java ME Connected Device Configuration (CDC) specification for non-graphical Java APIs for small electronic devices.
- PBP 1.1 [4] or later: the Java ME Personal Basis Profile (PBP) specifies a standards-based graphical user interface framework that support resource-constrained devices.

Taken together, the CDC and PBP provide a complete J2ME application environment for consumer products and embedded devices.

The following table is organized by four classes of GEM terminal specification targets: OTT, IPTV, broadcast and packaged media. For all but the OTT and IPTV targets, there are two profiles, the enhanced profile (abbreviated "E.P." in the table) and the interactive profile (abbreviated "I.P." in the table).

NOTE: The packaged media profile is present in the following table, but GEM 1.1 and GEM 1.2 changes have not been analyzed in the context of the packaged media profile.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 61: Platform profile definitions

Area	Specification	IPTV Target	OTT Target	Broadcast Targets		Packaged Media Targets	
				E.P.	I.P.	E.P.	I.P.
Static Formats							
Bitmap pictures	7.1.1.3, "PNG" + 15.1, "PNG - restrictions"	M	M	M	M	M	M
	7.1.1.3, "PNG" without restrictions	-	-	-	-	-	-
	7.1.1.4, "GIF"	-	-	-	-	-	-
	7.1.2, "MPEG-2 I-Frames"	O	-	M	M	M	M
	7.1.1.2, "JPEG" + 15.3, "JPEG - restrictions"	-	-	M	-	M	-
	7.1.1.2, "JPEG" without restrictions	M	M	-	M	-	M
Audio clips	7.1.4, "Monomedia format for audio clips"	M	M	M	M	M	M
Video drips	7.1.3, "MPEG-2 Video "drips""	O	-	M	M	M	M
Text encoding	7.1.5, "Monomedia format for text"	M	M	M	M	M	M
Media Streaming formats							
Video	7.1.5, "Monomedia format for text"	M	M	M	M	M	M
Audio	7.2.1, "Audio"	M	M	M	M	M	M
Subtitles	7.2.3, "Subtitles"	-	-	-	-	-	-
Container	7.2.4. “Containers”	O	M	-	-	-	-
Manifest	7.2.5 “Streaming Manifest”	O	O	-	-	-	-
Fonts							
Built in	Character set see annex E, "Character set" Metrics see annex D, "Text presentation" Face: UK RNIB "Tiresias"	O	O	M	M	O	O
Downloadable	7.4.1, "PFR"	O	O	M	M	O	O
	7.4.2, "OpenType"	O	O	O	O	O	O
Broadcast channel protocols							
MPEG-2	6.2.2, "MPEG-2 sections"	O	O	M	M	M	M
Object Carousel	6.2.5, "Object carousel"	O	O	M	M	-	-
IP Multicast	IP Multicast stack based on: 6.2.6, "Protocol for delivery of IP multicast over the broadcast channel" 6.2.7, "Internet Protocol (IP)" 6.2.8, "User Datagram Protocol (UDP)" 6.2.10, "IP signalling"	-	-	O	Ro	-	-
Interaction channel protocols							
TCP/IP	6.3.3, "Transmission Control Protocol" 6.3.2, "Internet Protocol"	M	M	-	M	-	M
UDP/IP	6.3.2, "Internet Protocol" 6.3.8, "User Datagram Protocol"	M	M	-	M	-	M
IGMP	6.5.1.5, “Internet Group Management Protocol (IGMP)”	O	O	-	-	-	-
RTP/UDP/IP	6.3.2, "Internet Protocol" 6.3.8, "User Datagram Protocol" 6.5.1.3, “Real Time Protocol (RTP)”	O	O	-	-	-	-
RTSP/UDP/IP	6.3.2, "Internet Protocol" 6.3.8, "User Datagram Protocol" 6.5.1.4, “Real Time Streaming Protocol (RTSP)”	O	O	-	-	-	-
HTTP	6.3.7.1, "HTTP 1.1"	O	M	-	O	-	O
	6.3.7.2, "GEM profile of HTTP 1.0"	M	-	-	M	-	M
DSMCC-UU RPC	6.3.4, "UNO-RPC" 6.3.5, "UNO-CDR" 6.3.6, "DSM-CC User to User"	-	-	-	O	-	-
DNS	6.3.9, "DNS"	M	M	-	M	-	M
HTTPS	6.3.7.3, "HTTPS"	M	M	-	M	-	M
Interaction Channel File System	6.4.1, "File system implemented only by the interaction channel"	O	O	-	M	-	M
DSMCC / HTTP hybrid	6.4.2, "Hybrid between broadcast stream and interaction channel"	O	O	-	M	-	M
IPTV	5, "Basic architecture"	O	O	O	O	O	O

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 277 of 861

Application Model							
Application Model	All parts of clause 9, "Application model" except those clauses (and their subclauses) identified below	M	M	M	M	M	M
	9.6.1, "Applications loaded from the interaction channel"	O	O	O	O	O	O
	9.6.2, "Stored services"	O	O	O	O	-	-
	9.7, "Lifecycle of internet access applications"	O	O	-	M	-	O
	9.9, "Stored and cached applications"	O	O	O	O	-	-
	9.13, "Unbound Applications"	O	O	O	O	O	O
Application Signalling							
Application Signalling	10, "Application signalling"	M	M	M	M	M	M
DVB-J							
DVB-J Platform	All parts of clause 11, "DVB-J platform" except those clauses (and their subclauses) identified below.	M	M	M	M	M	M
Presentation APIs	11.4.1, "Graphical User Interface API"	O	O	O	O	O	O
Data Access APIs	11.5.2, "Support for Multicast IP over the Broadcast Channel"	-	-	O	Ro	-	-
	11.5.3, "Support for IP over the Return Channel"	M	M	-	M	-	M
	11.5.4, "MPEG-2 Section Filter API"	O	O	M	M	O	O
	11.5.5, "Mid-Level Communications API" as modified by 11.5, "Data access APIs"	M	M	O	M	O	M
	11.5.7, "File Storage Device Access"	O	O	O	O	O	O
Service Information and Selection APIs	11.6.3, "Tuning API"	O	O	M	M	-	-
	11.6.4, "Conditional access API"	-	-	-	-	-	-
	11.6.6, "Service discovery and selection for IPTV"	O	O	-	-	-	-
	11.6.7, "Integration between protocol independent SI API and TV-Anytime"	O	O	-	-	-	-
Common Infrastructure APIs	11.7.4, "Basic MPEG concepts"	O	O	M	M	-	-
	11.7.6 "Content referencing"	O	O	O	O	O	O
	11.7.7, "Common error reporting"	O	O	M	M	M	M
	11.7.8, "Plug-in APIs"	O	O	M	M	O	O
	11.7.9, "Provider API"	O	O	-	-	-	-
	11.7.10, "Content referencing for IPTV"	O	O	-	-	-	-
	11.7.11, "TV-Anytime content referencing and metadata"	O	O	-	-	-	-
	11.7.12, “Content referencing for OTT”	O	M	-	-	-	-
Security	11.8.2, "APIs for return channel security"	M	M	-	M	-	M
	11.8.3, "Additional permissions classes"	O	O	M	M	-	-
	11.8.6, "DVB Extensions for Cryptography"	O	O	O	O	O	O
Other APIs	11.9.4, "Non-CA smart card API"	O	O	O	O	O	O
	11.9.5.2, "JDOM"	O	O	-	-	-	-
	11.9.6, "GEM terminal hardware API"	O	O	O	O	O	O
	11.9.7, “Content Download API”	O	O	-	-	-	-
	11.4.2.5.7, “Streaming Monitoring API”	O	M	-	-	-	-
	11.4.2.5.8, “Media Stream Synchronization API”	O	O	-	-	-	-
Content Referencing	11.11.1, "Transport stream"	O	O	M	M	O	O
	11.11.2, "Network"	O	O	M	M	O	O
	11.11.4.4, “Content referencing for IPTV”	M	O	-	-	-	-
	11.11.4.5, “Content referencing for OTT services”	O	M	-	-	-	-
	11.11.5, "Program event"	O	O	M	M	O	O
	11.11.9, "Drip feed decoder"	O	O	M	M	O	O
Stand-alone Apps	11.12.2, "Stored services"	O	O	O	O	O	O
OCAP APIs	11.15, "APIs defined in OCAP"	O	O	O	O	O	O
Internet Access	11.14, "Internet Access"	O	O	-	O	-	O
	17, "Internet access clients"	O	O	-	O	-	O

Key:

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Optional APIs:

An optional API means optional functionality, i.e. a GEM terminal is not required to include it.

Terminal implementations, which use eager linking VMs (see section 11.2. of the Java Language Specification [11]) shall provide API signatures with empty implementation (Compilation stubs) for all non-implemented optional APIs. This is required to prevent VM startup failures on these implementations.

For each package of an optional API, where only an empty implementation (stub) exists, there MUST be a system property consisting of the package name with a property value of "STUBBED". Additionally these classes shall contain a static initialization block which throws a NoClassDefFoundError to indicate an attempt to illegally use it.

NOTE: In this version of the specification, for IPTV the functional equivalent for clause 6.2.5, "Object carousel" is optional. This will be revisited in a future version of the present document. A mechanism for mounting a filesystem using `ServiceDomain.attach(Locator)` and a way of sending triggers that can be received using the DSMCCStreamEvent API are important for interoperability, e.g. with packaged media applications. A future version of the present document may make this level of functional equivalent mandatory, and IPTV implementations that wish to support interoperability with these commonly used capabilities have to support these functionalities.

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

15.1 PNG - restrictions

PNG is defined as in PNG Version 1.0 [43]; later versions of PNG introduce additional chunk types, which are not required to be supported. Receivers should ignore gAMA (gamma) and cHRM (chromaticity) chunks in PNG files.

GEM terminals are required to support all of the PNG colour types defined in PNG Specification Version 1.0 [43] (see table 62). GEM terminals are responsible for mapping these colours to those used by the terminal's OSD.

Any combination of PNGs with different colour types may be active at any one time. Similarly, terminals are responsible for mapping RGB16 direct colour specifications to colours that the OSD can support.

Table 62: PNG formats

Colour Type	Allowed Bit Depths	Interpretation
0	1, 2, 4, 8, 16	Each pixel is a grayscale sample.
2	8, 16 per component	Each pixel is an R,G,B triple.
3	1, 2, 4, 8	Each pixel is a palette index; PLTE chunk must appear.
4	8, 16	Each pixel is a grayscale sample, followed by an alpha sample.
6	8, 16	Each pixel is an R,G,B triple, followed by an alpha sample.

15.2 Minimum media formats supported by DVB-J APIs

Table 63 specifies the minimum set of media types that implementations of the "Enhanced Broadcast Profile" and "Interactive Broadcast Profile" shall support (see clause 4.1.6, "Profiles"). It also identifies the APIs that shall provide this support:

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 63: Media type support required in Enhanced and Interactive Broadcast profiles

Media type	Reference	API(s)
Downloadable fonts	7.4	`org.dvb.ui.FontFactory`
Audio from file	7.1.4	`org.havi.ui.HSound` JMF only with references to files
MPEG I frame images	7.1.2	`org.havi.ui.HBackgroundImage`
PNG images	7.1.1.3	`java.awt.Image`
JPEG images	7.1.1.2	`java.awt.Image`
DVB service	EN 300 468 [14]	JMF only with references to DVB services for playback direct from the network
Video "drips"	7.1.3	`org.dvb.media.DripFeedDataSource`
Text files	7.1.5	All Java APIs supporting reading or writing text files.
Subtitles	7.2.3	JMF only as part of a DVB service

Support for subtitles is optional.

Support of MPEG-2 Video "drips" is optional for the IPTV target.

Media formats in table 63, "Media type support required in Enhanced and Interactive Broadcast profiles" may be replaced with functional equivalents as permitted by clause 15.6, "Functional equivalents".

15.3 JPEG - restrictions

The restricted JPEG specification is as specified in clause 7.1.1.2, "JPEG" except that the "progressive DCT-based" mode is excluded.

15.4 Locale support

Support of resources for the following locales is required:

- One guaranteed Locale java.util.Locale.UK. (“en”. ”GB”).
- Zero (or more) implementation dependent ones.

Further it is guaranteed that the default Locale shall have resources. The default Locale is implementation dependent.

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

15.5 Video raster format dependencies

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

15.5.1 Standard Definition (PAL/SECAM or NTSC resolution)

15.5.1.1 Logical pixel resolution

The logical pixel resolution shall be 72 dots per inch.

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

15.6 Functional equivalents

Table 64 lists the set of mandatory and optional functional equivalents. Those elements marked mandatory shall be defined in a GEM terminal specification. Those marked optional may be defined in a GEM terminal specification.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Each functional equivalent is identified by a name in the leftmost column of table 64.

If a GEM terminal specification wishes to include technology that is not required by GEM, it may do so only for one or more named functional equivalents listed in column 1 in table 64 and shall be conformant with the clauses for that element referenced by table 64.

For each mandatory named functional equivalent, the GEM terminal specification shall explicitly indicate that it adopts the entire definition as specified in the current document or shall specify a replacement as required by the corresponding GEM clause(s) in table 64.

For each optional named functional equivalent, the GEM terminal specification shall explicitly indicate whether it is supported. If supported, the GEM terminal specification shall indicate that it adopts the entire definition as specified in the current document or shall specify a replacement as required by the corresponding GEM clause(s) in table 64.

Table 64 is organized by the three classes of GEM terminal specification targets, IPTV, broadcast and packaged media. For all but the IPTV target, there are two profiles, the enhanced profile (abbreviated "E.P." in the table) and the interactive profile (abbreviated "I.P." in the table).

Table 64: Functional equivalents

Name	GEM Clause(s)	IPTV Target	OTT Target	Broadcast Targets		Packaged Media Targets	
				E.P.	I.P.	E.P.	I.P.
Arch	5, "Basic architecture"	O	O	O	O	O	O
Carousel	6.2.5, "Object carousel" See also 11.7.2, "Application discovery and launching APIs"	O	O	M	M	M	M
IP MPE	6.2.6, "Protocol for delivery of IP multicast over the broadcast channel"	O	O	O	O	O	O
SI	6.2.9, "Service information" 11.6.1, "Signalling-specific service information API" Annex O, "Integration of the JavaTV SI API"	M	M	M	M	M	M
Broadcast IP signalling	6.2.10, "IP signalling"	O	O	O	O	O	O
IPTV Protocols	6.5, "IPTV protocols"	M	-	-	-	-	-
OTT Protocols	6.6, "OTT Protocols"	-	M	-	-	-	-
Audio	7.2.1, "Audio"	M	M	M	M	M	M
Video	7.2.2, "Video"	M	M	M	M	M	M
Subtitles	7.2.3, "Subtitles" 11.4, "Presentation APIs" - classes related to subtitles	O	O	O	O	O	O
Audio Clips	7.1.4, "Monomedia format for audio clips"	M	M	M*	M*	M	M
Resident Fonts	7.3, "Resident fonts"	M	M	M*	M*	O	O
Downloadable Fonts	7.4, "Downloadable fonts"	M*	M*	M*	M*	M	M
Application Signalling	10.2, "Program specific information" 10.4, "Application Description" 10.5, "DVB-J specific Application Description" 11.7.2, "Application discovery and launching APIs"	M	M	M	M	M	M
Application Authentication	12.2, "Authentication of applications" (see notes 1 and 2) 12.9.1, "Certificate Revocation Lists"	M	M	M	M	M	M
Conditional Access	11.4, "Presentation APIs," classes related to conditional access 11.6.4, "Conditional access API"	O	O	O	O	O	O

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 281 of 861

Name	GEM Clause(s)	IPTV Target	OTT Target	Broadcast Targets		Packaged Media Targets	
				E.P.	I.P.	E.P.	I.P.
Content Referencing	11.7.6, "Content referencing" 11.11.11, "Methods working on many locator types" 14.1, "Namespace mapping" 14.9, ″Content referencing for IPTV/ OTT″	M	M	M	M	M	M
Graphics Resolution	D.3.4.2, "Horizontal resolution" G.1.1, "Device capabilities" G.4, "Resident fonts and text rendering"	M	M	M	M	M	M
Text Wrapping	D.3.7.2, "Text wrapping setting is true"	M	M	M	M	M	M
RCMM	12.9.2, "Root certificate management"	M	M	M	M	M	M
Active Format Descriptor	N.1, "Active Format Definition"	O	O	O	O	O	O
NOTE 1: See also text in clause 12.6.2.6, "Credentials" and text in clause 12.9, "Certificate management". NOTE 2: See also text in clause 12.9.2, "Root certificate management".							
Key: M Functional equivalent required M* MHP definition of the functional equivalent required. O Optional feature, no functional equivalent required. - Not required/Not applicable.							

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

15.6.1 Modifications to MHP Definitions of Functional equivalents

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

15.6.1.1 Carousel

15.6.1.1.1 NSAP Address

The DVB Carousel NSAP Address structure may be modified by replacing the definition of `specifierType`, `specifierData`, and `dvb_service_location` structures with structures specified in the GEM terminal specification.

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

15.6.1.1.2 Content type descriptor

If a content type descriptor is not present in a file object, GEM terminal specifications may define another mechanism to determine the MIME type of a file that takes precedence over the mapping based on the extension portion of the file name, provided that a correctly built MHP object carousel will be processed in a way compatible with MHP.

15.6.1.1.3 Application Icons Descriptor

GEM terminal specifications using the functional equivalent of application signalling may define new application types. If they do, they may define icon locator semantics for the application types they introduce in the application icons descriptor.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

15.6.1.2 Application Signalling

15.6.1.2.1 Transport protocol descriptor

GEM terminal specifications using the functional equivalent named "application signalling" may define new `protocol_id`s and corresponding selector bytes provided that new `protocol_id` values shall be registered with the DVB.

15.6.1.2.2 AIT descriptor tag values

GEM terminal specifications that include the functional equivalent named "Application Signalling" and that define additional descriptors in the AIT shall register the descriptor tag values with the DVB.

NOTE: For information purposes current registrations are shown in clause 10.6, "Constant Values".

15.6.1.3 Application Name Descriptor

GEM terminal specifications using the functional equivalent named "application signalling" may modify the encoding of the `application_name_char` field of the application name descriptor provided that all strings that start with the value 0x20 are interpreted as ASCII strings, in accordance with annex A of EN 300 468 [14].

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

16 Registry of constants

16.1 System constants

Table 65: Registry of constants

Entity	Value	Description
PTimerMinRepeatInterval	40 ms	This (or optionally a smaller) value shall be returned by javax.tv.util.TVTimer.getMinRepeatInterval(). See clause 11.9.1, "Timer support".
PTimerGranularity	10ms	This (or optionally a smaller) value shall be returned by javax.tv.util.TVTimer.getGranularity(). See clause 11.9.1, "Timer support".

16.2 DVB-J constants

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

This clause to be populated with the values of public final static symbols from the various Java APIs.

16.2.1 Public and Protected final static primitive fields from DVB packages

The following is a list of the values assigned for public and protected final static primitive fields defined in the DVB defined DVB-J packages:

```
public final static int org.dvb.application.AppAttributes.DVB_J_application = 1;
public final static int org.dvb.application.AppAttributes.DVB_HTML_application = 2;
public final static int org.dvb.application.AppProxy.STARTED = 0;
public final static int org.dvb.application.AppProxy.DESTROYED = 1;
public final static int org.dvb.application.AppProxy.NOT_LOADED = 2;
public final static int org.dvb.application.AppProxy.PAUSED = 3;
public final static int org.dvb.application.AppsDatabaseEvent.NEW_DATABASE = 0;
public final static int org.dvb.application.AppsDatabaseEvent.APP_CHANGED = 1;
public final static int org.dvb.application.AppsDatabaseEvent.APP_ADDED = 2;
public final static int org.dvb.application.AppsDatabaseEvent.APP_DELETED = 3;
public final static int org.dvb.application.DVBJProxy.LOADED = 5;

public static final int org.dvb.application.DVBHTMLProxy.LOADING=6;
public static final int org.dvb.application.DVBHTMLProxy.KILLED=7;

public final static int org.dvb.dsmcc.DSMCCObject.FROM_CACHE = 1;
public final static int org.dvb.dsmcc.DSMCCObject.FROM_CACHE_OR_STREAM = 2;
public final static int org.dvb.dsmcc.DSMCCObject.FROM_STREAM_ONLY = 3;
public final static int org.dvb.dsmcc.DSMCCObject.FORCED_STATIC_CACHING=4;

public final static int org.dvb.event.UserEvent.UEF_KEY_EVENT = 1;
public final static int org.dvb.io.ixcIxcRegistry.SERVICE=1;
public final static int org.dvb.io.ixcIxcRegistry.PAGE=2;
public final static int org.dvb.io.ixcIxcRegistry.GLOBAL=3;

public final static int org.dvb.io.persistent.FileAttributes.PRIORITY_LOW = 1;
public final static int org.dvb.io.persistent.FileAttributes.PRIORITY_MEDIUM = 2;
public final static int org.dvb.io.persistent.FileAttributes.PRIORITY_HIGH = 3;

public final static int org.dvb.media.PresentationChangedEvent.STREAM_UNAVAILABLE = 0;
public final static int org.dvb.media.PresentationChangedEvent.CA_FAILURE = 1;
public final static int org.dvb.media.PresentationChangedEvent.CA_RETURNED = 2;
public final static int org.dvb.media.VideoFormatControl.ASPECT_RATIO_UNKNOWN = -1;
public final static int org.dvb.media.VideoFormatControl.AFD_NOT_PRESENT = -1;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_UNKNOWN = -1;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_NONE = 0;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_FULL = 1;
public final static int org.dvb.media.VideoFormatControl.DAR_4_3 = 1;
public final static int org.dvb.media.VideoFormatControl.ASPECT_RATIO_4_3 = 2;
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

```
public final static int org.dvb.media.VideoFormatControl.AFD_16_9_TOP = 2;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_LB_16_9 = 2;
public final static int org.dvb.media.VideoFormatControl.DAR_16_9 = 2;
public final static int org.dvb.media.VideoFormatControl.ASPECT_RATIO_16_9 = 3;
public final static int org.dvb.media.VideoFormatControl.AFD_14_9_TOP = 3;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_LB_14_9 = 3;
public final static int org.dvb.media.VideoFormatControl.ASPECT_RATIO_2_21_1 = 4;
public final static int org.dvb.media.VideoFormatControl.AFD_GT_16_9 = 4;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_CCO = 4;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_PAN_SCAN = 5;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_LB_2_21_1_ON_4_3 = 6;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_LB_2_21_1_ON_16_9 = 7;
public final static int org.dvb.media.VideoFormatControl.AFD_SAME = 8;
public final static int org.dvb.media.VideoFormatControl.DFC_PLATFORM = 8;
public final static int org.dvb.media.VideoFormatControl.AFD_4_3 = 9;
public final static int org.dvb.media.VideoFormatControl.AFD_16_9 = 10;
public final static int org.dvb.media.VideoFormatControl.AFD_14_9 = 11;
public final static int org.dvb.media.VideoFormatControl.AFD_4_3_SP_14_9 = 13;
public final static int org.dvb.media.VideoFormatControl.AFD_16_9_SP_14_9 = 14;
public final static int org.dvb.media.VideoFormatControl.AFD_16_9_SP_4_3 = 15;
public final static int org.dvb.media.VideoFormatControl.DFC_PROCESSING_16_9_ZOOM = 9;
public final static byte org.dvb.media.VideoPresentationControl.POS_CAP_OTHER = -1;
public final static byte org.dvb.media.VideoPresentationControl.POS_CAP_FULL = 0;
public final static byte
org.dvb.media.VideoPresentationControl.POS_CAP_FULL_IF_ENTIRE_VIDEO_ON_SCREEN = 1;
public final static byte org.dvb.media.VideoPresentationControl.POS_CAP_FULL_EVEN_LINES = 3;
public final static byte
org.dvb.media.VideoPresentationControl.POS_CAP_FULL_EVEN_LINES_IF_ENTIRE_VIDEO_ON_SCREEN = 4;

public final static int org.dvb.net.rc.RCInterface.TYPE_PSTN = 1;
public final static int org.dvb.net.rc.RCInterface.TYPE_ISDN = 2;
public final static int org.dvb.net.rc.RCInterface.TYPE_DECT = 3;
public final static int org.dvb.net.rc.RCInterface.TYPE_CATV = 4;
public final static int org.dvb.net.rc.RCInterface.TYPE_LMDS = 5;
public final static int org.dvb.net.rc.RCInterface.TYPE_MATV = 6;
public final static int org.dvb.net.rc.RCInterface.TYPE_RCS = 7;
public final static int org.dvb.net.rc.RCInterface.TYPE_UNKNOWN = 8;
public final static int org.dvb.net.rc.RCInterface.TYPE_OTHER = 9;

public final static short org.dvb.si.DescriptorTag.NETWORK_NAME = 64;
public final static short org.dvb.si.DescriptorTag.SERVICE_LIST = 65;
public final static short org.dvb.si.DescriptorTag.STUFFING = 66;
public final static short org.dvb.si.DescriptorTag.SATELLITE_DELIVERY_SYSTEM = 67;
public final static short org.dvb.si.DescriptorTag.CABLE_DELIVERY_SYSTEM = 68;
public final static short org.dvb.si.DescriptorTag.BOUQUET_NAME = 71;
public final static short org.dvb.si.DescriptorTag.SERVICE = 72;
public final static short org.dvb.si.DescriptorTag.COUNTRY_AVAILABILITY = 73;
public final static short org.dvb.si.DescriptorTag.LINKAGE = 74;
public final static short org.dvb.si.DescriptorTag.NVOD_REFERENCE = 75;
public final static short org.dvb.si.DescriptorTag.TIME_SHIFTED_SERVICE = 76;
public final static short org.dvb.si.DescriptorTag.SHORT_EVENT = 77;
public final static short org.dvb.si.DescriptorTag.EXTENDED_EVENT = 78;
public final static short org.dvb.si.DescriptorTag.TIME_SHIFTED_EVENT = 79;
public final static short org.dvb.si.DescriptorTag.COMPONENT = 80;
public final static short org.dvb.si.DescriptorTag.MOSAIC = 81;
public final static short org.dvb.si.DescriptorTag.STREAM_IDENTIFIER = 82;
public final static short org.dvb.si.DescriptorTag.CA_IDENTIFIER = 83;
public final static short org.dvb.si.DescriptorTag.CONTENT = 84;
public final static short org.dvb.si.DescriptorTag.PARENTAL_RATING = 85;
public final static short org.dvb.si.DescriptorTag.TELETEXT = 86;
public final static short org.dvb.si.DescriptorTag.TELEPHONE = 87;
public final static short org.dvb.si.DescriptorTag.LOCAL_TIME_OFFSET = 88;
public final static short org.dvb.si.DescriptorTag.SUBTITLING = 89;
public final static short org.dvb.si.DescriptorTag.TERRESTRIAL_DELIVERY_SYSTEM = 90;
public final static short org.dvb.si.DescriptorTag.MULTILINGUAL_NETWORK_NAME = 91;
public final static short org.dvb.si.DescriptorTag.MULTILINGUAL_BOUQUET_NAME = 92;
public final static short org.dvb.si.DescriptorTag.MULTILINGUAL_SERVICE_NAME = 93;
public final static short org.dvb.si.DescriptorTag.MULTILINGUAL_COMPONENT = 94;
public final static short org.dvb.si.DescriptorTag.PRIVATE_DATA_SPECIFIER = 95;
public final static short org.dvb.si.DescriptorTag.SERVICE_MOVE = 96;
public final static short org.dvb.si.DescriptorTag.SHORT_SMOOTHING_BUFFER = 97;
public final static short org.dvb.si.DescriptorTag.FREQUENCY_LIST = 98;
public final static short org.dvb.si.DescriptorTag.PARTIAL_TRANSPORT_STREAM = 99;
public final static short org.dvb.si.DescriptorTag.DATA_BROADCAST = 100;
public final static byte org.dvb.si.PMTStreamType.MPEG1_VIDEO = 1;
public final static byte org.dvb.si.PMTStreamType.MPEG2_VIDEO = 2;
public final static byte org.dvb.si.PMTStreamType.MPEG1_AUDIO = 3;
public final static byte org.dvb.si.PMTStreamType.MPEG2_AUDIO = 4;
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

```
public final static short org.dvb.si.SIInformation.FROM_CACHE_ONLY = 0;
public final static short org.dvb.si.SIInformation.FROM_CACHE_OR_STREAM = 1;
public final static short org.dvb.si.SIInformation.FROM_STREAM_ONLY = 2;
public final static byte org.dvb.si.SIMonitoringType.NETWORK = 1;
public final static byte org.dvb.si.SIMonitoringType.BOUQUET = 2;
public final static byte org.dvb.si.SIMonitoringType.SERVICE = 3;
public final static byte org.dvb.si.SIMonitoringType.PMT_SERVICE = 4;
public final static byte org.dvb.si.SIMonitoringType.PRESENT_FOLLOWING_EVENT = 5;
public final static byte org.dvb.si.SIMonitoringType.SCHEDULED_EVENT = 6;
public final static byte org.dvb.si.SIRunningStatus.UNDEFINED = 0;
public final static byte org.dvb.si.SIRunningStatus.NOT_RUNNING = 1;
public final static byte org.dvb.si.SIRunningStatus.STARTS_IN_A_FEW_SECONDS = 2;
public final static byte org.dvb.si.SIRunningStatus.PAUSING = 3;
public final static byte org.dvb.si.SIRunningStatus.RUNNING = 4;
public final static short org.dvb.si.SIServiceType.UNKNOWN = -1;
public final static short org.dvb.si.SIServiceType.DIGITAL_TELEVISION = 1;
public final static short org.dvb.si.SIServiceType.DIGITAL_RADIO_SOUND = 2;
public final static short org.dvb.si.SIServiceType.TELETEXT = 3;
public final static short org.dvb.si.SIServiceType.NVOD_REFERENCE = 4;
public final static short org.dvb.si.SIServiceType.NVOD_TIME_SHIFTED = 5;
public final static short org.dvb.si.SIServiceType.MOSAIC = 6;
public final static short org.dvb.si.SIServiceType.PAL = 7;
public final static short org.dvb.si.SIServiceType.SECAM = 8;
public final static short org.dvb.si.SIServiceType.D_D2_MAC = 9;
public final static short org.dvb.si.SIServiceType.FM_RADIO = 10;
public final static short org.dvb.si.SIServiceType.NTSC = 11;
public final static short org.dvb.si.SIServiceType.DATA_BROADCAST = 12;
public final static short org.dvb.si.SIServiceType.MHP_APPLICATION = 16;

public final static int org.dvb.test.DVBTest.UNTESTED = -5;
public final static int org.dvb.test.DVBTest.UNRESOLVED = -4;
public final static int org.dvb.test.DVBTest.HUMAN_INTERVENTION = -3;
public final static int org.dvb.test.DVBTest.OPTION_UNSUPPORTED = -2;
public final static int org.dvb.test.DVBTest.FAIL = -1;
public final static int org.dvb.test.DVBTest.PASS = 0;

public final static int org.dvb.ui.DVBAlphaComposite.CLEAR = 1;
public final static int org.dvb.ui.DVBAlphaComposite.SRC = 2;
public final static int org.dvb.ui.DVBAlphaComposite.SRC_OVER = 3;
public final static int org.dvb.ui.DVBAlphaComposite.DST_OVER = 4;
public final static int org.dvb.ui.DVBAlphaComposite.SRC_IN = 5;
public final static int org.dvb.ui.DVBAlphaComposite.DST_IN = 6;
public final static int org.dvb.ui.DVBAlphaComposite.SRC_OUT = 7;
public final static int org.dvb.ui.DVBAlphaComposite.DST_OUT = 8;
public final static int org.dvb.ui.DVBBufferedImage.TYPE_ADVANCED = 20;
public final static int org.dvb.ui.DVBBufferedImage.TYPE_BASE = 21;
public final static int org.dvb.ui.DVBTextLayoutManager.HORIZONTAL_START_ALIGN = 1;
public final static int org.dvb.ui.DVBTextLayoutManager.HORIZONTAL_END_ALIGN = 2;
public final static int org.dvb.ui.DVBTextLayoutManager.HORIZONTAL_CENTER = 3;
public final static int org.dvb.ui.DVBTextLayoutManager.VERTICAL_START_ALIGN = 4;
public final static int org.dvb.ui.DVBTextLayoutManager.VERTICAL_END_ALIGN = 5;
public final static int org.dvb.ui.DVBTextLayoutManager.VERTICAL_CENTER = 6;
public final static int org.dvb.ui.DVBTextLayoutManager.LINE_ORIENTATION_HORIZONTAL = 10;
public final static int org.dvb.ui.DVBTextLayoutManager.LINE_ORIENTATION_VERTICAL = 11;
public final static int org.dvb.ui.DVBTextLayoutManager.START_CORNER_UPPER_LEFT = 20;
public final static int org.dvb.ui.DVBTextLayoutManager.START_CORNER_UPPER_RIGHT = 21;
public final static int org.dvb.ui.DVBTextLayoutManager.START_CORNER_LOWER_LEFT = 22;
public final static int org.dvb.ui.DVBTextLayoutManager.START_CORNER_LOWER_RIGHT = 23;

public static final int org.dvb.internet.InternetServiceFilter.EMAIL_CLIENT = 1;
public static final int org.dvb.internet.InternetServiceFilter.WWW_CLIENT = 2;
public static final int org.dvb.internet.InternetServiceFilter.NEWS_CLIENT = 3;
public static final int org.dvb.application.AppProxy.INVALID = 8;

public static final int org.dvb.spi.ict.ResourceTransportObject.LOAD_TYPE_FILE = 1;
public static final int org.dvb.spi.ict.ResourceTransportObject.LOAD_TYPE_MEMORY= 0;
public static final int org.dvb.smartcard.SmartCardReaderEvent.SMART_CARD_ERROR = 3;
public static final int org.dvb.smartcard.SmartCardReaderEvent.SMART_CARD_IN = 0;
public static final int org.dvb.smartcard.SmartCardReaderEvent.SMART_CARD_MUTED = 2;
public static final int org.dvb.smartcard.SmartCardReaderEvent.SMART_CARD_OUT = 1;
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

17 Internet access clients

This feature is optional in all profiles of GEM.

17.1 Referencing DVB services and content within WWW content

GEM internet access clients shall support the following mechanisms of referencing DVB services and content from within WWW content.

- Use of a reference to a DVB service in an HTML "href" shall launch that DVB service in the same way as is done by GEM applications using the GEM service selection API or the end user of the GEM terminal using the navigator to select DVB services. GEM internet access clients are not covered by the GEM security model and shall be able to perform service selection and tuning without any consideration of permissions. The normal GEM resource management mechanisms shall be applied if tuning is required and the tuner concerned is reserved.

NOTE: Only GEM applications with `SelectPermission` can start an internet access client. Such applications can in any case use the service selection API to also select broadcast services. Hence this does not weaken the GEM security model.

GEM applications shall only be started in response to selection of a reference to a DVB service. All GEM applications run in the scope of a DVB service and cannot run on their own outside of a service of some type.

17.2 Minimum requirements for internet clients

Table 66 defines some minimum network protocols which shall be supported by those internet client applications which form part of the GEM internet access profile.

Table 66

Internet Client Application	Minimum required protocols
WWW browser	HTTP (see clause 6.3.7.2, "GEM profile of HTTP 1.0") HTTPS (see clause 6.3.7.3, "HTTPS")
Email client	SMTP for sending email (see RFC 821 [71]) or HTTP to a WebMail server. Protocols for receiving email are implementation dependent.
Usenet news	NNTP (RFC 977 [72]) or HTTP to a WebNews server.

WWW browser internet clients shall support use of the "mailto" URL as defined in RFC 2818 [73] and RFC 2368 [74].

Email internet clients shall support use of the "http" and "https" URLs as defined in section 3.2.2 of RFC 2616 [45].

If a Usenet news internet client is present, then the WWW browser and E-mail internet clients shall also support the use of the "news" URL as defined by RFC 1738 [61]. Usenet news internet clients, if present, shall support the use of the "http", "https", and "mailto" URLs.

17.3 Internet streamed media

There is no requirement for GEM terminals to support internet streaming media content however where these are both supported and visible to DVB-J applications, it shall be done using the Java Media Framework (Java TV [16]).

17.4 Internet connection management

Internet client applications shall always use the default ISP configured into the GEM application. For example, in a GEM terminal with only one `RCInterface` that is a `ConnectionRCInterface`, this is the same default as defined by

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 287 of 861

`ConnectionRCInterface.setTargetToDefault`. If this default ISP is reached via a `ConnectionRCInterface` that is currently connected to another target, the normal GEM resource management mechanism shall apply with the internet client application having the resource priority of the GEM application that launched it. If the internet client application fails to reserve such a `ConnectionRCInterface`, the service selection shall fail with a `SelectionFailedEvent` with reason code `INSUFFICIENT_RESOURCES`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

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

A.1 Void

A.2 Errata to DAVIC

The following errata to DAVIC [17] shall apply.

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

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

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

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

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

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

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

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

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

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

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

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

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

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

Shall be considered to be replaced with:

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

A.2.3 org.davic.mpeg

A.2.3.1 General

The following classes are considered to have a no-argument protected constructor:

- `ElementaryStream`.
- `Service`.
- `TransportStream`.

A.2.3.2 NotAuthorizedException

The specification is considered to include `org.davic.mpeg.NotAuthorizedException` as specified below:

org.davic.mpeg

NotAuthorizedException

Syntax

```
public class NotAuthorizedException extends java.lang.Exception implements
org.davic.mpeg.NotAuthorizedInterface
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.davic.mpeg.NotAuthorizedException
```

All Implemented Interfaces:

```
NotAuthorizedInterface, java.io.Serializable
```

Description

This class is thrown by MPEG related APIs when access is requested to information which is scrambled and to which access is not permitted by the security system.

Constructors:

NotAuthorizedException()

```
    public NotAuthorizedException()
```

Constructs a NotAuthorizedException with no detail message.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 290 of 861

NotAuthorizedException(String)

```
public NotAuthorizedException(java.lang.String s)
```

Constructs a NotAuthorizedException with the specified detail message.

Parameters:

`s` - the detail message.

Methods:

getElementaryStreams()

```
public ElementaryStream[] getElementaryStreams()
```

If getType() returns ELEMENTARY_STREAM, then this method returns the set of ElementaryStreams that could not be descrambled. Otherwise it returns null.

Specified by:

`NotAuthorizedInterface.getElementaryStreams()` in interface `NotAuthorizedInterface`

Returns:

either the set of ElementaryStreams that could not be descrambled or null.

getReason(int)

```
public int[] getReason(int index)
```

Returns the reason(s) why descrambling was not possible.

Specified by:

`NotAuthorizedInterface.getReason(int)` in interface `NotAuthorizedInterface`

Parameters:

`index` - If the component to which access failed is a Service, index shall be 0. Otherwise index shall refer to one stream in the set returnedby getElementaryStreams().

Returns:

an array of length 2 where the first element of the array is the major reason and the second element of the array is the minor reason.

Throws:

`IndexOutOfBoundsException` - If the component to which access failed is a Service, this exception will be thrown if index is non zero. If the component(s) to which access failed was a (set of) elementary streams then this exception will be thrown where index is beyond the size of the array returned by getElementaryStreams.

See also:

`NotAuthorizedInterface.getElementaryStreams()`

getService()

```
public Service getService()
```

If getType() returns SERVICE, then this method returns the Service that could not be descrambled. Otherwise it returns null.

Specified by:

`NotAuthorizedInterface.getService()` in interface `NotAuthorizedInterface`

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 291 of 861

Returns:

either the Service that could not be descrambled or null.

getType()

```
public int getType()
```

Specified by:

`NotAuthorizedInterface.getType()` in interface `NotAuthorizedInterface`

Returns:

SERVICE or ELEMENTARY_STREAM to indicate that either a service (MPEG program) or one or more elementary streams could not be descrambled.

A.2.4 Chapter 9, "Application Format"

A.2.4.1 Section 9.4.7, "The MPEG-2 Section Filter API"

In the description of this class there are 2 instances of a cross reference to DAVIC part 10, section 115.3. In each case this shall be considered as a reference to DAVIC part 10, section 12.5.3.

A.2.5 org.davic.mpeg.sections

A.2.5.1 RingSectionFilter

Is considered to have the following text appended to its description:

- All sections in a ring section filter are initialized to empty when the ring section filter is first created. Clearing them to empty any time after this is the responsibility of the application. Starting a ring section filter shall not clear any of the sections to empty.

A.2.5.2 Section

A.2.5.2.1 clone()

Section is considered to have the method clone() with the following behaviour.

- A cloned Section object is a new and separate object. It is unaffected by changes in the state of the original Section object or restarting of the SectionFilter the source Section object originated from. The clone method must be implemented without declaring exceptions.

A.2.5.2.2 getData()

Remove the following text from the methods of `org.davic.mpeg.sections.Section`:

- (everything after the length field, not including a CRC check).

A.2.5.2.3 getFullStatus()

Is considered to have the following text appended to its description:

- Returns true when the `Section` object contains valid data.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.2.5.2.4 Class Description

The following text shall be considered to be present at the end of the class description:

- In this class, the term "section data" shall be considered to include the section header, any possible CRC and all bytes in between.

A.2.5.3 SectionFilter

A.2.5.3.1 Cross reference error

In the description of this class there are 12 instances of a cross reference to H7 of DAVIC [17]. In each case this shall be considered as a reference to E.8.1.

A.2.5.3.2 startFiltering(all signatures)

In these methods, the description of when the `FilterResourceException` shall be thrown is considered to have the following text appended to its description:

- This shall be applied whether the parent section filter group is connected to a TS or not, or whether this SectionFilter object is already started or not.

A.2.5.3.3 startFiltering(java.lang.Object, int, int, int, byte[], byte[])

Is considered to have the following text appended to its description:

- `IllegalFilterDefinitionException` is thrown if offset is too small.

A.2.5.3.4 startFiltering (appData, pid, tableId) exceptions

Like other `startFiltering` methods `org.davic.mpeg.sections.SectionFilter.startFiltering (appData, pid, tableId)` shall throw an `IllegalFilterDefinitionException` where:

- the Java integer is negative;
- the Java integer is larger than what is allowed for PID or table_id according to the relevant MPEG specification.

A.2.5.3.5 Started Section Filters

The class description is considered to have the following text added at its end.

- When a SectionFilterGroup is detached, either by the client or through resource withdrawal, started SectionFilters shall remain started. Hence if the SectionFilterGroup is re-attached, those filters shall re-activate.

A.2.5.4 SectionFilterGroup

A.2.5.4.1 attach

A `NotAuthorizedException` is added to the definition of this method.

A.2.5.4.2 Constructors

The constructors are considered to have.

- Throws `IllegalArgumentException` if `numberOfFilters` <1.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 293 of 861

A.2.5.4.3 sectionSize

All methods with a `sectionSize` parameter are considered to have the following text appended to their descriptions:

- Throws `IllegalArgumentException` if `sectionSize` <1.

A.2.5.4.4 newRingSectionFilter

Is considered to have the following text appended to its description:

- Throws `IllegalArgumentException` if `ringSize` <1.

A.2.5.4.5 Constructor(int, boolean)

Is considered to have the following text appended to its description:

- The scope of the `resourcePriority` shall be a single application only.

A.2.5.4.6 General

In the methods of this class, all occurrences of "(successful startFiltering)" shall be considered to read "(successful call to the startFiltering method of the newly created instance)".

A.2.5.5 TimeOutEvent

The specification is considered to include `org.davic.mpeg.sections.TimeOutEvent` as specified below:

org.davic.mpeg.sections
TimeOutEvent

Declaration

```
public class TimeOutEvent extends org.davic.mpeg.sections.EndOfFilteringEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.davic.mpeg.sections.org.davic.mpeg.sections.SectionFilterEvent
              |
              +--org.davic.mpeg.sections.org.davic.mpeg.sections.EndOfFilteringEvent
                    |
                    +--org.davic.mpeg.sections.TimeOutEvent
```

All Implemented Interfaces:

`java.io`.Serializable

Description

This event is generated if section filter operations time out within the period specified by the setTimeOut() method. For a SimpleSectionFilter it will be generated if no sections arrive within the specified period. For a TableSectionFilter, it will be generated if the complete table does not arrive within the specified time. For a RingSectionFilter, it will be generated if the specified time has elapsed since the arrival of the last section being successfully filtered.

Constructors:

TimeOutEvent(SectionFilter, Object)

```
public TimeOutEvent(SectionFilter f, java.lang.Object appData)
```

This constructs an TimeOutEvent event for the specified SectionFilter object.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Parameters:

`f` - the SectionFilter object which timed out.

`appData` - application data that was passed to the startFiltering method.

Methods:

getSource()

```
public java.lang.Object getSource()
```

This returns the SectionFilter object which timed out.

Overrides:

`EndOfFilteringEvent.getSource()` in class `EndOfFilteringEvent`

A.2.6 Simple Section Filter

The description of the getSection method shall be considered to have the following text added:

- This method shall return null if no section has been successfully filtered for this instance and the owning SectionFilterGroup is not attached.

A.2.7 org.davic.media

A.2.7.1 FreezeControl.resume()

Add the following to the description of the semantics for this method:

- If the player is started and if decoding of the media stream is not frozen then calls to this method shall have no effect.
- If the player is not started then the exception shall be thrown.

A.2.7.2 MediaTimePositionChangedEvent

Add the following constructor:

```
MediaTimePositionChangedEvent (
    Controller from,
    int previous,
    int current,
    int target,
    Time mediaTime)
```

With the following definition of parameters:

Parameters:

- from - the controller whose media position was changed;
- previous - the state the controller was in before this event;
- current - the state the controller was in at the time the event was generated;
- target - the state that the controller is heading to;
- mediaTime - the media time after the change.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.2.7.3 NotAuthorizedMediaException

The following constructors are considered to be removed from the specification:

```
NotAuthorizedMediaException()
NotAuthorizedMediaException(java.lang.String reason)
```

The following constructors are considered to be a normative part of the specification:

```
NotAuthorizedMediaException( org.davic.mpeg.Service, int reason )
NotAuthorizedMediaException( org.davic.mpeg.ElementaryStream[], int reason[] )
NotAuthorizedMediaException( org.davic.mpeg.ElementaryStream[], int[], int[] )
NotAuthorizedMediaException(Service, int major_reason,int minor_reason)
```

NotAuthorizedMediaException(ElementaryStream[], int[])

```
public NotAuthorizedMediaException(org.davic.mpeg.ElementaryStream[] e, int[] reason)
```

Constructor for exception due to failure accessing one or more MPEG elementary streams The caller of this constructor is responsible for ensuring the two arrays provided as parameters are the same size. The implementation is not expected to check this. The exception has no detail message.

Parameters:

`e` - the elementary streams which could not be accessed

`reason` - the reason why the exception was thrown for each elementary stream.

Use of the constructor NotAuthorizedMediaException(ElementaryStream[] e, int[] reason) will result in the major reason for each elementary stream being the one specified in the reason parameter to the method and the minor reason being OTHER as defined in NotAuthorizedInterface.

NotAuthorizedMediaException(Service, int)

```
public NotAuthorizedMediaException(org.davic.mpeg.Service s, int reason)
```

Constructor for exception due to failure accessing an MPEG service. The exception has no detail message.

Parameters:

`s` - the service which could not be accessed.

`reason` - the reason why the service could not be accessed.

Use of the constructor NotAuthorizedMediaException(Service, int reason) will result in the major reason for the service being the one specified in the reason parameter to the method and the minor reason being OTHER as defined in NotAuthorizedInterface.

NotAuthorizedMediaException(ElementaryStream[], int[], int[])

```
public NotAuthorizedMediaException(ElementaryStream[] e, int[] major_reason,int[]
minor_reason)
```

Constructor for exception due to failure accessing one or more MPEG elementary streams The caller of this constructor is responsible for ensuring the three arrays provided as parameters are the same size. The implementation is not expected to check this.

Parameters:

`e` - the elementary streams which could not be accessed.

`major_reason` - the major reason why the exception was thrown for each elementary stream.

`minor_reason` - the minor reason why the exception was thrown for each elementary stream.

NotAuthorizedMediaException(Service, int, int)

```
public NotAuthorizedMediaException(org.davic.mpeg.Service s, int major_reason,
int minor_reason)
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Constructor for exception due to failure accessing an MPEG service.

Parameters:

`s` - the service which could not be accessed.

`major_reason` - the major reason why the service could not be accessed.

`minor_reason` - the minor reason why the service could not be accessed.

Since:

MHP 1.0.2

A.2.7.4 LanguageControl

A.2.7.4.1 Class description

The following description of semantics is considered to be present:

- If more than one stream with the same language exists, the behaviour of `selectLanguage(String)` is to select the first listed in the network signalling.

NOTE: This is equivalent to item b under clause 11.4.2.3, "Default media player behaviour".

In the methods `getCurrentLanguage` and `selectDefaultLanguage`, the following additional text shall be considered to be present:

- If no content of the appropriate media type for the control is present, a String of length zero is returned.

In the method `selectDefaultLanguage`, the following shall be considered to be appended to the method description:

- If this information is not available, a String of length zero is returned.

A.2.7.4.2 selectLanguage(String)

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

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

A.2.8 org.davic.net

A.2.8.1 InvalidLocatorException

The following class definition is considered to be a normative part of the specification:

org.davic.net

InvalidLocatorException

Syntax

```
public class InvalidLocatorException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.davic.net.InvalidLocatorException
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

All Implemented Interfaces:

java.io.Serializable

Description

This exception is thrown when one or more parameters to construct a Locator are invalid.

Constructors:

InvalidLocatorException()

```
public InvalidLocatorException()
```

Constructor without reason.

InvalidLocatorException(String)

```
public InvalidLocatorException(java.lang.String reason)
```

Constructor for the exception with a specified reason.

Parameters:

`reason` - the reason why the exception was raised.

A.2.8.2 Locator

A.2.8.2.1 Locator()

The no-argument constructor for `org.davic.net.Locator` is considered to not be present. The absence of any description on the method indicates that this was an editing error in the DAVIC specification.

A.2.8.2.2 toExternalForm()

Is considered to have the following text appended to its description:

- If the instance of `Locator` has been created using `Locator(java.lang.String url)` and the URL is a non-null invalid URL the behaviour is implementation dependent.

A.2.8.3 tuning

A.2.8.3.1 NetworkInterfaceController

A.2.8.3.1.1 reserve()

The semantic of the following throws clause is corrected as follows:

```
Throws:     NoFreeInterfaceException
            raised if the requested network interface can not be reserved
```

The following from the semantic of this method:

- If this NetworkInterfaceController has already reserved another NetworkInterface, then it will either release that NetworkInterface and reserve the specified one, or throw an exception. If the specified NetworkInterface has already been reserved by this NetworkInterfaceController, then this method does nothing.

is replaced with the following:

- If this NetworkInterfaceController has currently reserved another NetworkInterface, then it will either release that NetworkInterface and reserve an appropriate one, or throw an exception. If a NetworkInterface that is able to tune to the specified transport stream is currently reserved by this NetworkInterfaceController, then this method does nothing.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 298 of 861

A.2.8.3.1.2 reserveFor()

The following from the semantic of this method:

- If this NetworkInterfaceController has already reserved another NetworkInterface, then it will either release that NetworkInterface and reserve an appropriate one, or throw an exception. If NetworkInterfaceController has already reserved a NetworkInterface that is able to tune to the specified transport stream, then this method does nothing.

is replaced with the following:

- If this NetworkInterfaceController has currently reserved another NetworkInterface, then it will either release that NetworkInterface and reserve an appropriate one, or throw an exception. If a NetworkInterface that is able to tune to the specified transport stream is currently reserved by this NetworkInterfaceController, then this method does nothing.

A.2.8.3.1.3 tune()

Replace "this NetworkInterface" with "the NetworkInterface reserved by this NetworkInterfaceController" in section H.5.4.3 of DAVIC specification.

A.2.8.3.2 NetworkInterface

A.2.8.3.2.1 Protected constructor

This class is considered to have a no argument protected constructor with the following statement attached to it:

- This constructor is provided for the use of implementations and specifications which extend this specification. Applications shall not define sub-classes of this class. Implementations are not required to behave correctly if any such application defined sub-classes are used.

A.2.9 org.davic.net.tuning

A.2.9.1 Figure H-1

In figure H-1, "Tuning API object model: Base classes", in the class `NetworkInterface`, the method `getURL()` shall be considered to be "`getLocator()`".

A.2.9.2 Figure H-2

In:

- figure H-2: "Tuning API object model: Exceptions".

The exception `IncorrectURLException` shall be considered to be absent.

A.2.9.3 Network Interface

The following additional methods are considered to be present:

```
/**
 * Tests for equality where two NetworkInterface objects are equal if and only if
 * they control the same physical tuner.
 * @param other the reference object with which to compare.
 * @return true if and only if other is an instance of NetworkInterface and calls to other
 * control the same physical tuner as this.
 */
public boolean equals(Object other)

/**
 * Returns a hash code value for the object. which obeys the contract of Object.hashCode().
 * @return a hash code value for the object.
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 299 of 861

```
 */
public int hashCode()
```

A.2.10 Extensibility and Over-Riding

The DAVIC specification DAVIC 1.4.1p9 [17] is considered to include the following;

- The addition of public or protected constructors, methods or fields to the `org.davic` packages is allowed except where this would cause a compliant MHP application to fail. Examples of the latter include the adding of abstract methods to classes or adding of methods to interfaces which application classes will implement.
- Within the `org.davic` package and its sub packages, overriding inherited public and protected methods not specified as being overridden is an allowable implementation option.

NOTE: This language does not apply to constructors or to fields, because inherited fields and constructors cannot be overridden in the Java language.

A.3 Additional GEM requirements on Java TV

A.3.1 javax.tv.util.TVTimerSpec

NOTE: Java TV 1.1 [16] requires that the argument to deschedule(TVTimerSpec) be the instance passed as argument to the method TVTimer.schedule. Implementations of GEM terminal specifications based on Java TV 1.1 are required to obey this JavaTV-defined behavior. This may be implemented in a way consistent with the GEM requirements by implementing TVTimer.schedule such that it always returns the instance passed in as argument. Note that Java TV's specification of TVTimer.schedule allows the implementation to modify the instance passed in as argument, e.g. to account for timer granularity.

A.3.2 javax.media.Manager

A.3.2.1 getDataSourceList()

In the JavaDoc for javax.media.Manager.getDataSourceList() the following text:

The first name in the list will always be:

```
media.protocol.<protocol>DataSource
```

shall be considered to be replaced with:

The first name in the list will always be:

```
media.protocol.<protocol>.DataSource
```

A.3.2.2 getHandlerClassList()

In the JavaDoc for javax.media.Manager.getHandlerClassList() the following text:

```
public static java.util.Vector getHandlerClassList(java.lang.String contentName)
```

Build a list of `Handler/CODE>` classes from the `content-prefix-list` and a `content` name.

The first name in the list will always be:

```
media.content.<contentName>.Handler
```

Each additional name looks like:

```
<content-prefix>.media.content.<contentName>.Player
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

for every <content-prefix> in the content-prefix-list.

Parameters:

`contentName` - The content type to use in the class name.

Returns:

A vector of strings where each one is a `Player` class-name.

shall be considered to be replaced with:

```
public static java.util.Vector getHandlerClassList(java.lang.String contentName)
```

Build a list of `Handler` classes from the `content-prefix-list` and a `content` name.

The first name in the list will always be:

```
media.content.<contentName>.Handler
```

Each additional name looks like:

```
<content-prefix>.media.content.<contentName>.Handler
```

for every `<content-prefix>` in the content-prefix-list.

Parameters:

`contentName` - The content type to use in the class name.

Returns:

A vector of strings where each one is a `Player` class-name.

A.4 HAVi

With reference to HAVi [50].

A.4.1 Drafting conventions

The HAVi specification does not follow the ETSI drafting rules as specified in SR 001 262 [i.10]. In particular, the term "should" shall be interpreted in terms of the ETSI drafting rules as being closer to "shall" than "should".

A.4.2 General

A.4.2.1 Thread-safety

As with the `java.awt package`, the `org.havi.ui` package is not required to be thread safe.

A.4.2.2 javadoc errors

The javadoc for the HAVi specification appears to be compiled against a version of the "java.*" packages more recent than that required for the MHP specification. All methods, fields and inner classes shown as inherited from classes outside the "org.havi" package namespace which are not present in the MHP specification for that class shall be considered to be absent from the HAVi specification.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.3 Event mechanism

This clause presents a clarification for the HAVi event mechanism.

A.4.3.1 Introduction

One of the reasons for the HAVi event mechanism is to provide a mechanism for widgets to request a type of input that may not be available on the input device supported by a platform. A HAVi widget can indicate its input preference by implementing one or more `HxxxInputPreferred` interfaces. This is important to enable application provided widgets to have access to the same capabilities as have always been available to platform provided ones.

An `HxxxInputPreferred` interface indicates the widget that implements the interface expects a certain type of input events. E.g. on systems with limited input devices the platform can supply a virtual keyboard to generate the interface-specific events.

The `Hxxxable` and `HxxxValue` interfaces group methods common to various `HVisible` subclasses; they do not imply additional support from the platform.

The `HxxxInputPreferred` interfaces all imply possible support by a (virtual) keyboard. The platform shall provide a way for the user to generate the input expected by the focused component. This could mean automatically presenting some kind of virtual keyboard when a focused component requires input that could not be generated in another way. The platform can determine which virtual keys a widget requires by looking at the `HxxxInputPreferred` interfaces it implements.

For `keyCode` input, `HNavigationInputPreferred.getNavigationkeys()` and `HKeyboardInputPreferred.getValidInput()` allow the platform to determine which `keyCodes` the particular widgets expect. Platform implementations shall provide the user with a means to generate the `keyCodes` expected by all `HxxxInputPreferred` interfaces implemented by the widget having input focus (with the exception of `keyCodes` not supported according to `org.havi.ui.event.HKeyCapabilities`).

HAVi events generated by the platform and sent to a `HComponent` are `HActionEvent`, `HAdjustmentEvent`, `HItemEvent`, `HTextEvent`, `HRcEvent`, `HFocusEvent` and `HKeyEvent`.

A.4.3.2 Overview of HAVi events

`HComponents` indicate they want to receive `HFocusEvents` by implementing the `HNavigationInputPreferred` interface. For `HNavigationInputPreferred`, the `keyCodes` for navigation keystrokes generated on the `HNavigationInputPreferred` will be passed to the `HNavigationInputPreferred` as an `HFocusEvent` `transferId`. The `HNavigationInputPreferred` will process an `HFocusEvent` of id `FOCUS_TRANSFER` in its `processHFocusEvent` method by calling `requestFocus` on the `HNavigable` associated with the `keyCode` returned by `HFocusEvent.getTransferId()`, if there is one.

Implementations of `HNavigationInputPreferred` will not send this `HFocusEvent` to `FocusListeners` or `HFocusListeners`, it is only used by the platform to communicate the transferId `to` the `HNavigationInputPreferred`.

The manner in which the keystrokes are generated is implementation-specific, as some platforms may supply a more extensive hardware keyboard, while other platforms may choose to provide a virtual keyboard to generate these keystrokes.

Also, HAVi allows generation of the HAVi events on top of AWT (in `HComponent.processEvent`) from other events as an implementation option, to allow for platforms unable to generate HAVi event directly. As a result of this, it is platform specific whether `processEvent` is ever called with a HAVi event as argument, or that the event is generated in this method from other events. In such an implementation, a widget may receive the original, unexpected, platform-specific, non-HAVi `java.awt` events as well as the translated HAVi events. To avoid confusion, a HAVi widget is supposed to process only the HAVi events and applications should ignore other events (i.e. not add listeners for `java.awt` events and not use them through `processXxxEvent` methods).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 302 of 861

For the handling of navigation keys, an implementation of the HNavigationInputPreferred interface receives the keycodes for navigation as the transferId of org.havi.ui.event.HFocusEvent, and not as a java.awt.event.KeyEvent or org.havi.ui.event.HKeyEvent. (On platforms which generate HAVi events from other events, it is possible that the component does receive a java.awt.event.KeyEvent, however this is platform-specific and applications should not rely on this).

Hcomponent.processEvent is responsible to dispatch HFocusEvents to the processHFocusEvent method; this method will call requestFocus on the target of a transferId for a FOCUS_TRANSFER event, and send HFocusEvents of type FOCUS_GAINED and FOCUS_LOST to any registered HFocusListeners for classes that support adding these listeners. Depending on which other HAVi interfaces the component implements and on whether it extends HVisible, this method will also change the state of the widget, play appropriate sounds, and notify any listeners specified in these other interfaces for a focus change.

An HComponent implementing HKeyboardInputPreferred will receive input in the form of HKeyEvents. On platforms with limited input devices, the platform can present a virtual keyboard for the user to generate the requested keyCodes. The platform can use the input type of the HKeyboardInputPreferred to determine which keys to present on this virtual keyboard, however, the platform is not required to prevent the user from generating more keyCodes than those requested by the HKeyboardInputPreferred.

HAVi allows to generate the HKeyEvents from other events in HComponent.processEvent, so any KeyEvents received by the component which are not HKeyEvents should be ignored as platform-specific.

Hcomponent.processEvent is responsible for dispatching HKeyEvents to the processHKeyEvent method; this method will send the HKeyEvent to any registered HKeyListeners for classes that support adding these listeners.

If an HComponent implements both HNavigationInputPreferred and HKeyboardInputPreferred, the platform will provide a way for the user to generate the keyCodes for both HFocusEvents and HKeyEvents in an unambiguous way (e.g. while in edit mode, keyCodes are converted into HKeyEvents, otherwise they generate HFocusEvents). The getEditMode method can be called by the platform to determine the current input mode of the widget. The platform will provide a platform specific way to switch between any different input modes as required by any limitations of its input devices, and notify the widget of this change by sending it an appropriate START_CHANGE or END_CHANGE event.

If the same keyCode is expected for different input types (e.g. VK_ENTER is set as navigation code while the widget also expects it as an HKeyEvent), then the platform will present the user with means to separately generate each of the HAVi events for that keyCode (in the case of the example, an HFocusEvent with transferId VK_ENTER and an HKeyEvent). Application authors should realize that to use keyCodes which are likely to have multiple functions may force some platforms to present a less than perfectly user-friendly display.

For platforms that generate HAVi events in HComponent.processEvent, extra events may be received in addition to the HAVi events. For example, for HFocusEvent FOCUS_GAINED and FOCUS_LOST, if the HFocusEvent is generated directly, processEvent should send this HFocusEvent to both processHFocusEvent and to processFocusEvent (the second one will happen in the superclass method). If the platform generates HAVi events in processEvent, when a FocusEvent is received by HComponent.processEvent, it will be sent to processFocusEvent and a new HFocusEvent will be created to send to processHFocusEvent.

For the other HAVi events, like HActionEvent, HAdjustmentEvent, HItemEvent and HTextEvent events, the behaviour is analogous. The platform will provide a way for the user to generate the events, either directly or from other events (translated in HComponent.processEvent), on HComponents that implement the HActionInputPreferred, HAdjustmentInputPreferred, HSelectionInputPreferred, and HKeyboardInputPreferred interfaces, respectively.

Implementations of the interfaces do not have to override processEvent so as to send these events to the appropriate processHxxEvent method, this will be the responsibility of Hcomponent.processEvent. The processHxxEvent method will call any registered listeners and perform any other specified actions, such as changing states, playing sounds, etc.

HAdjustmentInputPreferred and HSelectionInputPreferred, which extend HOrientable, allow the platform to select a way for the user to generate HAdjustmentEvents and HItemEvents which best matches the component's representation.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 303 of 861

A.4.3.3 Relation between HAVi events and AWT events

Applications which extend from `java.awt.Component` (and not `org.havi.ui.HComponent`) and which register event listeners from the `java.awt.event` package shall receive the behaviour defined for the `java.awt.event` package. If events from the `org.havi.ui.event` package are generated on such a `Component`, these shall be sent to any registered listeners according to the behaviour defined for their parent class in `java.awt.event`.

HAVi allows implementations that generate `HxxxEvents` from other events, so an `HComponent` may receive the `HxxxEvent` in addition to those other events. For example, you could get both `havi.ui.event.HKeyEvents` sent to `HKeyListeners` and `awt.event.KeyEvents` sent to AWT KeyListeners.

For subclasses that extend from `Component` and not from `HComponent`, it is implementation dependent whether they receive the events defined in `awt.event` or a subclass of these events defined in havi (e.g. `org.havi.ui.event.HKeyEvent` instead of `java.awt.event.KeyEvent`). However, in the latter case the behaviour will be as defined for the parent class, e.g. an `org,havi.ui.event.HKeyEvent` will go to `KeyListeners` added with `Component.addKeyListener`. Also, applications cannot assume that such `Components` will be able to receive `KeyEvents` with `keyCodes` other than those specified in the MHP minimum profile.

A.4.3.4 Application guidelines

Any implementation of a `HxxxInputPreferred` interface shall implement the processing of the `HxxxEvent` and calling of listeners in the `processHxxEvent` method (where this event processing is not inherited from a parent class). It is the platforms' responsibility to provide a way in which the user can generate the events requested by the widget. Since the HAVi events may be generated in the `processEvent` method on some platforms, interface implementers should not assume that the `processEvent` method will receive the HAVi events on every platform. On platforms where the HAVi events are generated in `processEvent`, it is the platform's responsibility to call the `processHxxEvent` method with the HAVi event.

A.4.4 org.havi.ui

A.4.4.1 HActionable

The class `HListGroup` is considered to be removed from the list of "Platform Classes".

A.4.4.1.1 getActionSound

The description of this method shall be considered to be replaced with the following

- Return the last action sound set by the `setActionSound` method or null if no action sound has been set.

A.4.4.1.2 Class description

The following sentence immediately before the heading "Interaction States":

- HAVi action events are discussed in detail in the HActionInputPreferred interface description.

Shall be modified to read:

- HAVi action events are discussed in detail in the HActionEvent interface description.

A.4.4.1.3 setActionCommand

The parameters clause for the "command" parameter shall be considered to be extended with the following sentence:

- To remove the command specify a null command parameter.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.2 HActionInputPreferred

The description of this method shall be considered to have the following text added:

- If this `HActionInputPreferred` has no action command then an empty string shall be returned.

A.4.4.3 HAdjustmentInputPreferred

A.4.4.3.1 Interface description

In the interface description of `HAdjustmentInputPreferred` after the sentence:

- For platforms with a restricted number of physical keys this may involve a "virtual keyboard" or similar mechanism.

is considered to be extended by the following text:

- The system might use the information returned by the method `getOrientation()` of the super interface to select appropriate key mappings for this event. The mechanisms to generate this event shall not be effective while the component is disabled (see `HComponent.setEnabled()`).

Also, the following text:

- All interoperable implementations of the `HAdjustmentInputPreferred` interface must extend `HComponent`.

Is considered to be replaced with:

- Widgets of HAVi compliant applications implementing the `HAdjustmentInputPreferred` interface must have `HComponent` in their inheritance tree.

Also, in the following text:

- Note that the `java.awt.Component` method `isFocusTraversable` should always return true for a `java.awt.Component` implementing this interface.

The word "should" is considered to be replaced with "shall".

A.4.4.3.2 getAdjustMode

The last sentence of the description of `getAdjustMode()` is considered to be extended with:

- Note that these events are ignored, if the component is disabled. See also: `HComponent.setEnabled()`.

A.4.4.3.3 proccessHAdjustmentEvent

The description of `proccessHAdjustmentEvent()` is considered to be extended with:

- Widgets implementing this interface shall ignore `HAdjustmentEvents`, while the component is disabled. See also: `HComponent.setEnabled()`.

A.4.4.3.4 setAdjustMode

The description of `setAdjustMode()` is considered to be extended with:

- Calls to this method shall be ignored, if the component is disabled. See also: `HComponent.setEnabled()`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 305 of 861

A.4.4.4 HAdjustmentValue

A.4.4.4.1 Class description

The class `HListGroup` is considered to be removed from the list of "Platform Classes".

A.4.4.4.2 getAdjustmentSound

This method is considered to have the following text added to the method description.

- null shall be returned if this method is called before its corresponding set method.

A.4.4.4.3 getBlockIncrement

This method is considered to have the following text added to the method description.

- 1 shall be returned if this method is called before its corresponding set method.

A.4.4.4.4 getUnitIncrement

This method is considered to have the following text added to the method description.

- 1 shall be returned if this method is called before its corresponding set method.

A.4.4.5 HAnimateEffect

A.4.4.5.1 getRepeatCount

This method shall be considered to have the following text added to the method description.

- Except for `HAnimateEffect` implementations that specify a different default, `getRepeatCount()` returns `REPEAT_INFINITE` if no call to `setRepeatCount()` has previously been made.

A.4.4.6 HBackgroundDevice

A.4.4.6.1 setBackgroundConfiguration

In the `setBackgroundConfiguration` method, the sentence below shall be considered to be added to the description of this method:

- Applications can prevent or limit changes to configurations of other, not intended, devices by using constants ZERO_GRAPHICS_IMPACT, ZERO_VIDEO_IMPACT and ZERO_BACKGROUND_IMPACT in their configuration templates.

The following shall be added as the first item in the bulleted list:

- If an application tries to select a configuration which is not valid for that device at that time or when the device is in a particular mode then an HConfigurationException shall be thrown.

A.4.4.6.2 getConfigurations

The following shall be added to the end of the method description.

- The set of configurations returned may include ones which are only valid for the device at particular times or when the device is in a particular mode.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.6.3 Constructor

In the constructor of this class, the following sentence:

- It is not intended that applications should directly construct `HBackgroundDevice` objects.

Shall be considered to be replaced with the following:

- An interoperable application shall not subclass the `HBackgroundDevice` class.

A.4.4.7 HBackgroundImage

A.4.4.7.1 Constructors - HBackgroundImage(String), HBackgroundImage(URL)

The sentence:

- Loading of the data for the object is not required at this time.

shall be considered to be replaced with:

- Loading of the data for the object shall not happen at this time.

A.4.4.7.2 load

The following text is considered to be added to the description of this method:

- Multiple calls to load shall each add an extra listener, all of which are informed when the loading is completed. If load is called with the same listener more than once, the listener shall then receive multiple copies of a single event.

A.4.4.7.3 Constructor(byte[])

In the following sentence:

- If the byte array does not contain a valid image then this constructor shall throw a java.lang.IllegalArgumentException.

"shall" shall be considered to say "may".

A.4.4.8 HComponent

A.4.4.8.1 processEvent

The following text is a clarification of the HAVi specification:

The implementation of the method `HComponent.processEvent()` shall ensure that key events which are translated to HAVi events shall not be reported to `processKeyEvent()` or reported to `KeyListeners`. Key events which are not translated to HAVi events shall be reported to `processKeyEvent()` and `KeyListeners` as defined in the Java specification.

NOTE: If applications override `processEvent` they may terminally disturb these processes. Applications should not do this without extreme care, as the results may be very implementation dependent.

A.4.4.9 HComponentOrdering

A.4.4.9.1 addAfter, addBefore

In the returns clause, the text:

- is successfully added.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 307 of 861

shall be considered to read:

- is successfully added or was already present.

A.4.4.10 HEventMulticaster

A.4.4.10.1 Class description

The following text is considered to be added to the description of this class:

- The `HEventMulticaster` class is intended to assist platform or subclass implementers with the handling of HAVi events. Implementations are not required to use this class to dispatch HAVi events. Applications should not extend the `HEventMulticaster` class and implementations are not required to behave correctly if an application does extend this class. If an extended multicaster is desired, `AWTEventMulticaster` should be used rather than `HEventMulticaster`.

Also, extend the following paragraph in the class description:

- It is an implementation option for this class to insert other classes in the inheritance tree (for example `java.awt.AWTEventMulticaster`). It is allowed that this may result in `HEventMulticaster` inheriting additional methods beyond those specified here.

with:

- If this class does extend `java.awt.AWTEventMulticaster`, it is allowed for the fields defined in this class to be inherited from that parent class.

A.4.4.10.2 Constructor

The following text is considered to form part of the constructor description:

- The parameters a and b passed to the constructor shall be used to populate the fields a and b of the instance.

A.4.4.10.3 remove

In the description of this method, replace:

- returns the resulting multicast listener;

with:

- returns the result.

A.4.4.11 HFontCapabilities

A.4.4.11.1 getSupportedCharacterRanges

The returns clause of this method shall be considered to be extended with the following text:

- including where the capabilities of the font are unknown.

The following paragraph in the method description:

- Support for a character range does not imply that ALL characters within that range are available in the specified font.

Shall be considered to be extended with the following extra sentence:

- Support does require that at lease one character from the range is included in the font.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.12 HGraphicsDevice

A.4.4.12.1 getBestConfiguration

The following text shall be considered to be added to the end of the second paragraph of the method description:

- If there are such equally best configurations, the one which is returned by this method is an implementation dependent selection from among those which are equally best.

A.4.4.12.2 setGraphicsConfiguration

In the `setGraphicsConfiguration` method, the sentence below:

- Applications can prevent or limit changes to configurations of other, not intended, devices by using constants `ZERO_GRAPHICS_IMPACT` and `ZERO_VIDEO_IMPACT` in their configuration templates.

Shall be considered to be extended as follows:

- Applications can prevent or limit changes to configurations of other, not intended, devices by using constants `ZERO_GRAPHICS_IMPACT`, `ZERO_VIDEO_IMPACT` and `ZERO_BACKGROUND_IMPACT` in their configuration templates.

The following shall be added as the first item in the bulleted list:

- If an application tries to select a configuration which is not valid for that device at that time or when the device is in a particular mode then an HConfigurationException shall be thrown.

A.4.4.12.3 getConfigurations

The following shall be added to the end of the method description.

- The set of configurations returned may include ones which are only valid for the device at particular times or when the device is in a particular mode.

A.4.4.12.4 Constructor

In the constructor of this class, the following sentence:

- It is not intended that applications should directly construct `HGraphicsDevice` objects.

Shall be considered to be replaced with the following:

- An interoperable application shall not subclass the `HGraphicsDevice` class.

A.4.4.13 HGraphicsConfigTemplate

A.4.4.13.1 setPreference

The following text shall be added to the third paragraph of the method description:

- Calling this method with null for the object parameter shall have no effect if the preference is not currently set in the template.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.14 HListElement

A.4.4.14.1 Class description

The class description is considered to be extended by the following:

- The methods `setIcon()` and `setLabel()` of `HListElement` shall not be used for elements, which are part of `HListGroup`. If an application requires to alter the content, it shall either replace the entire element, or remove it temporarily and re-add it after the content was changed.

A.4.4.15 HListGroup

A.4.4.15.1 Class description

In the following text:

- The `HListGroup` is a user interface component representing a list of selectable items (`HListElements`) which contain static read-only graphical and / or textual content.

The phrase "static read-only" is considered to be removed.

Also, extend the following text:

- Interoperable HAVi applications shall not add `HListElement` more than once. If an application requires items with identical contents (label and/or icon), then additional items shall be created. The behaviour of the `HListGroup` if duplicates are added is implementation specific.

With:

- The methods `setIcon()` and `setLabel()` of `HListElement` shall not be used for elements, which are part of `HListGroup`. If an application requires to alter the content, it shall either replace the entire element, or remove it temporarily and re-add it after the content was changed.

Also, in the text:

- the minimum size is the size to present one element or an implementation specific minimum (32 x 32 for example) if no elements are present.

The phrase "if no elements are present" is considered to be replaced with:

- if label and icon size are not set (see `HListGroupLook.getMinimumSize()`).

Also, under the parameters table under heading "Default parameter values exposed in the constructors" the description of the parameter "items" missing is considered to read:

- The initial list of elements for this `HListGroup` or `null` for an empty list.

A.4.4.15.2 Fields

A.4.4.15.2.1 ITEM_NOT_FOUND

In the description:

- A constant which may be returned from `getIndex` if the requested element is not found in the content.

The word "may" is considered to be replaced by "shall".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.15.2.2 ADD_INDEX_END

In the description:

- A constant for use with `addItem` and `addItems` which specifies that the new items should be appended to the end of the list.

The word "should" is considered to be replaced by "shall".

A.4.4.15.3 Use of "action" and "actioning"

The use of the words "action" and "actioning" in the following methods in `HListGroup` does not imply any connection between `HListGroup` and `HActionable`. There is no such connection in the API:

- `getCurrentIndex()`
- `getCurrentItem()`
- `setCurrentItem()`
- `getSelectionIndices()`
- `getSelection()`

A.4.4.15.4 addItem(HListItem item, int index)

In the description of parameter "index" all occurrences of the word "items" is considered to be replaced by "item".

A.4.4.15.5 addItem(HListItem items[], int index)

In the parameter description:

- item - the item to add.

is considered to be replaced by:

- items - the items to add.

A.4.4.15.6 getIconSize

The description of this method is considered to be extended by:

- If label and icon size do not match the size per element, the associated `HListGroupLook` is allowed to use other sizes during the rendering process. This size shall be used by `HListGroupLook` to calculate the size per element.

A.4.4.15.7 getLabelSize

The description of this method is considered to be extended by:

- If label and icon size do not match the size per element, the associated `HListGroupLook` is allowed to use other sizes during the rendering process. This size shall be used by `HListGroupLook` to calculate the size per element.

A.4.4.15.8 getOrientation

The following text:

- The orientation controls how an associated `HLook` lays out the component and affects the visual behaviour of the `HAdjustmentEvent` and `HItemEvent` events. `HListGroups` do not receive `HAdjustmentEvents`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Is considered to be replaced by:

- The orientation controls how an associated `HLook` lays out the component and affects the visual behaviour of `HItemEvent` events.

A.4.4.15.9 setFocusTraversal

In all cases the phrase "then null should be specified" is considered to be replaced by "then null shall be specified".

A.4.4.15.10 setItemSelected

The following text:

- If a successful call to this method causes the selection to change an HItemEvent shall be sent to any registered listeners. If the selection does not change then no HItemEvent shall be sent.

Shall be considered to be replaced with:

- If a call to this method causes an item to become deselected an HItemEvent with an ID of ITEM_CLEARED shall be sent to all registered listeners. This can happen because either sel is false or this HListGroup is not in multi selection mode.
- If a call to this method causes a non selected item to become a selected item then an HItemEvent with an ID of ITEM_SELECTED shall be sent to all registered listeners.

A.4.4.15.11 setIconSize

The description of this method is considered to be extended by:

- If label and icon size do not match the size per element, the associated `HListGroupLook` is allowed to use other sizes during the rendering process. This size shall be used by `HListGroupLook` to calculate the size per element.

In the description of the size parameter, the following text:

- Dimension(DEFAULT_ICON_SIZE, DEFAULT_ICON_SIZE).

is considered to be replaced by

- Dimension(DEFAULT_ICON_WIDTH, DEFAULT_ICON_HEIGHT).

A.4.4.15.12 setLabelSize

The description of this method is considered to be extended by:

- If label and icon size do not match the size per element, the associated `HListGroupLook` is allowed to use other sizes during the rendering process. This size shall be used by `HListGroupLook` to calculate the size per element.

A.4.4.15.13 setListContent

The sentence:

- Any existing selection is discarded (which may cause an HItemEvent to be generated).

Is considered to be replaced by

- If any elements are selected, then the selection is discarded and an HItemEvent is generated.

The following text:

- Set the list content for this HListGroup. If any elements are selected, then the selection is discarded and an HItemEvent is generated.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Shall be considered to be replaced with:

- Set the list content for this HListGroup. If any items are selected, then the selection shall be discarded and an HItemEvent with an ID of ITEM_SELECTION_CLEARED shall be generated and sent to all registered listeners.
- If elements is null then the current active item index shall be set to ITEM_NOT_FOUND. If elements is non null then the current active item index shall be set to 0. An HItemEvent with an ID of ITEM_SET_CURRENT shall be sent to all registered listeners.

A.4.4.15.14 setMove

In all cases the phrase "then null should be specified" is considered to be replaced by "then null shall be specified".

A.4.4.15.15 setScrollPosition

This method is considered to have the following text added:

Parameters:

- scroll - the scroll position.

A.4.4.15.16 setSelectionMode

The parameter in the method signature called `adjust` shall be considered to be called `edit` as used in the parameter list.

A.4.4.15.17 removeItem

The following text:

- Remove the HListElement at the specified index. The item is also removed from the selection, if any is set. If this was the last item in the selection the entire selection is destroyed and calls to getSelection shall return null until new content and selections are created.
- If the act of removing an item causes the current active item index to change, an HItemEvent shall be sent.
- If the act of removing an item causes the selection to change, an HItemEvent shall be sent.

Shall be considered to be replaced with the following text:

- Removes the HListElement at the specified index. All following items are shifted.
- If the item is the only HListElement in this HListGroup then the current active item index shall be set to ITEM_NOT_FOUND. If the removal of the item causes a change in the current active item index then an HItemEvent with an ID of ITEM_SET_CURRENT shall be generated and sent to all registered listeners.
- If the item is selected then it shall be removed from the selection and an HItemEvent with an ID of ITEM_CLEARED shall be generated and sent to all registered listeners.

A.4.4.15.18 removeAllItems

The following text:

- Remove all the content. The selection is also destroyed and calls to getSelection shall return null until new content and selections are created.

Shall be considered to be replaced with:

- Removes all the content. If any items are selected, then the selection shall be discarded and an HItemEvent with an ID of ITEM_SELECTION_CLEARED shall be generated and sent to all registered listeners.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- The current active item index shall be set to ITEM_NOT_FOUND and an HItemEvent with an ID of ITEM_SET_CURRENT shall be generated and sent to all registered listeners.

A.4.4.15.19 setCurrentItem

The following paragraph shall be considered to form part of the description of this method.

- If index is valid for this HListGroup then the current active item index shall be set to index. If this causes a change in the current active item index then an HItemEvent with an ID of ITEM_SET_CURRENT shall be generated and sent to all registered listeners.

Under the Returns subsection, the following paragraph:

- true if the current item was changed, false if index was not a valid index for this HListGroup or the current item was not changed because it is already selected. No exception is thrown if index is not valid.

Shall be considered to be replaced with:

- true if the current item was changed, false if index was not a valid index for this HListGroup or the current item was not changed because it is already the current item. No exception is thrown if index is not valid.

A.4.4.15.20 setMultiSelection

The following text:

- Note that if the HListGroup is switched out of multiple selection mode and more than one item is selected, the selection shall change so that the first of the items is selected and the others are deselected. This will cause an HItemEvent to be sent to any registered listeners.

Shall be considered to be replaced with:

- Note that if the HListGroup is switched out of multiple selection mode and more than one item is selected, the selection shall change so that the first of the items is selected and the others are deselected. An HItemEvent with an ID of ITEM_CLEARED shall be sent to all registered listeners for each deselected item.

A.4.4.16 HListGroupLook

A.4.4.16.1 getMaximumSize

The description of `getMaximumSize()` is considered to be replaced by the following:

- Returns the size to present all elements of the specified `HVisible` plus any additional dimensions that the `HListGroupLook` requires for border decoration etc. If no elements are present, a dimension object is returned with width and height set to `java.lang.Short.MAX_VALUE`.

- The extra space required for border decoration can be determined from the `getInsets()` and `getElementInsets()` methods. The behaviour is not defined for the case, when a subclass overrides these methods. Application developers shall not assume any influence on the returned dimensions.

- The size per element shall be determined by calls to `getIconSize()` and `getLabelSize()` of `HListGroup`. If any of the values requests a default as specified by `DEFAULT_ICON_WIDTH`, `DEFAULT_ICON_HEIGHT`, `DEFAULT_LABEL_WIDTH` and `DEFAULT_LABEL_HEIGHT`, then an implementation specific default is used for the corresponding value(s).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Parameters:

- visible HVisible to which this HLook is attached.

  Returns:

  - A dimension object indicating this HListGroupLook's maximum size. See also:

    ```
    HListGroup.setIconSize()
    HListGroup.setLabelSize()
    HVisible.getMaximumSize()
    ```

A.4.4.16.2 getMinimumSize

The description of getMinimumSize() is considered to be replaced by the following:

- Returns the size to present one element of the specified HVisible plus any additional dimensions that the HListGroupLook requires for border decoration, etc.
- The extra space required for border decoration can be determined from the getInsets() and getElementInsets() methods. The behaviour is not defined for the case, when a subclass overrides these methods. Application developers shall not assume any influence on the returned dimensions.
- The size per element shall be determined by calls to getIconSize() and getLabelSize() of HListGroup. If any of the dimensions requests a default as specified by DEFAULT_ICON_WIDTH, DEFAULT_ICON_HEIGHT, DEFAULT_LABEL_WIDTH and DEFAULT_LABEL_HEIGHT, then an implementation specific default is used for the corresponding value(s).

Parameters:

- visible HVisible to which this HLook is attached.

  Returns:

  - A dimension object indicating this HListGroupLook's minimum size. See also:

    ```
    HListGroup.setIconSize()
    HListGroup.setLabelSize()
    HVisible.getMinimumSize()
    ```

A.4.4.16.3 getPreferredSize

The description of getPreferredSize() is considered to be replaced by the following:

- Gets the preferred size of the HVisible component when drawn with this HListGroupLook.
- If a default size for width and height was set with HVisible.setDefaultSize(), then the dimensions are rounded down to the nearest element (minimum of one) according to the orientation of the associated HListGroup, and any dimensions for border decorations etc. are added.
- If no default size was set or only for one dimension (i.e. height is NO_DEFAULT_HEIGHT or width is NO_DEFAULT_WIDTH), then the unset dimension(s) shall be sufficiently large to present five elements according to the HListGroup's orientation. Any dimensions for border decoration etc. are added.
- The extra space required for border decoration can be determined from the getInsets() and getElementInsets() methods. The behaviour is not defined for the case, when a subclass overrides these methods. Application developers shall not assume any influence on the returned dimensions.
- The size per element shall be determined by calls to getIconSize() and getLabelSize() of HListGroup. If any of the values requests a default as specified by DEFAULT_ICON_WIDTH, DEFAULT_ICON_HEIGHT, DEFAULT_LABEL_WIDTH and DEFAULT_LABEL_HEIGHT, then an implementation specific default is used for the corresponding value(s).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Parameters:

- visible `HVisible` to which this `HListGroupLook` is attached.

  Returns:

  - A dimension object indicating the preferred size of the `HVisible` when drawn with this `HListGroupLook`. See also:

    ```
    HListGroup.setIconSize()
    HListGroup.setLabelSize()
    HVisible.getPreferredSize()
    HVisible.setDefaultSize()
    ```

A.4.4.16.4 getValue

The first paragraph of `getValue()` is considered to be extended by the following:

- A non-null value represents the scroll position of the associated `HListGroup`. The value shall never be less than zero.

And the description of returns is considered to be replaced by the following:

- the non-negative scroll position associated with the specified pointer position or null

A.4.4.16.5 hitTest

In the description of `hitTest()` after

- then this method will return the index of that element.

is considered to be extended by the following:

- The `HListGroup` shall interpret this index as current item. If the value is `ADJUST_THUMB`, then the caller shall use `getValue()` to retrieve the actual scroll position corresponding to the specified pointer position.

And the returns description considered to be extended by the following:

- or a non-negative element index.

A.4.4.16.6 showLook

The description of `showLook()` is considered to be replaced by the following:

- This method is responsible for repainting the entire visible including the list content set on the `HListGroup`, and the visible background, subject to the clipping rectangle of the graphics object passed to it.
- If the method modifies the clipping rectangle of the graphics object, it shall restore the original rectangle upon return.
- `showLook()` paints the visible with its current background colour according to the `getBackgroundMode()` method of `HVisible` and draws any (implementation-specific) borders, regardless whether content is available or not. Note that by default the background mode is set to not paint a background. Furthermore on platforms which support transparent colours the background colour may be partially or completely transparent.
- Any resources explicitly associated with this look shall be loaded by it during its creation. Note that the "standard" looks do not load content by default.
- This method is called from the `paint()` method of `HVisible` and must never be called from elsewhere. Components wishing to redraw themselves shall call their repaint method in the usual way or call `widgetChanged()` under certain circumstances.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 316 of 861

- The labels of the associated HListElements shall be rendered by using the current text layout manager of the HListGroup. For each visible label is the render() method of HTextLayoutManager called. The position and size per label are specified as insets relatively to the bounds of HListGroup. Note that the bounds are independent of any borders of the HListGroup, but the insets have to include the borders per element, if any. The look shall use the method getLabelSize() of HListGroup to determine the size for each label. If the returned dimension object has DEFAULT_LABEL_WIDTH for the width and/or DEFAULT_LABEL_HEIGHT for the height as values, then this method shall use implementation specific value(s) as default(s) for the missing dimension(s) instead. If getTextLayoutManager() returns null, then labels shall not be rendered.

- If supported, scaling of icons shall reflect the resize mode of the visible within the area of the respective list element. The look shall use the method getIconSize() of HListGroup to determine the size for each icon. If the returned dimension object has DEFAULT_ICON_WIDTH for the width and/or DEFAULT_ICON_HEIGHT for the height as values, then this method shall use implementation specific value(s) as default(s) for the missing dimension(s) instead.

- Except for the alignment of labels and sizes of labels and icons, it is explicitly not defined, how this look arranges icons and labels within the elements' areas. Additionally, it is an implementation option to render labels and icons in other sizes than specified, if the available size per element is smaller or larger than label and icon size. It is also not defined, how the look presents the current item and selected items, or the current selection mode. The elements shall be layed out as specified by getOrientation() of the associated HListGroup.

- When the associated HListGroup contains more elements than presentable, the look shall make the user aware of that condition, e.g. by displaying an additional scrollbar reflecting the current scroll position. Again, the visible means by which this information is conveyed is not defined and implementation dependent. It is an implementation option for HListGroupLook to draw elements before the scroll position, in order to fill the available space.

- The behaviour of this method, when a subclass overrides the methods getInsets() or getElementInsets(), is not defined. Application developers shall not assume that the corresponding borders will appear as specified.

  Parameters:

  - g the graphics context.
  - visible the visible.
  - state the state parameter indicates the state of the visible, allowing the look to render the appropriate content for that state. Note that the default behaviour of HListGroupLook is to ignore this parameter, since the content of HListGroup is not state based. See also:

```
java.awt.Component.getBounds()
java.awt.Graphics.getClipBounds()
HListGroup.getCurrentItem()
HListGroup.getListContent()
HListGroup.getOrientation()
HListGroup.getScrollPosition()
HListGroup.setIconSize()
HListGroup.setLabelSize()
HListGroup.isItemSelected()
HTextLayoutManager.render()
```

A.4.4.16.7 Class description

The following sentence shall be considered to be removed.

- Borders are not drawn around the content.

A.4.4.16.8 showLook and renderVisible

The five paragraphs in showLook starting with "The labels of the associated HListElements shall be rendered..." and ending "...Application developers shall not assume that the corresponding borders will appear as specified." shall be considered to form part of the renderVisible method instead.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 317 of 861

A.4.4.16.9 renderVisible

The following shall be considered to be added to the end of this method description:

- The org.havi.ui.HExtendedLook.renderVisible method is responsible for painting any implementation specific borders for each HListElement as well as drawing of an additional scrollbar if required.

A.4.4.17 HLook

A.4.4.17.1 General

In the description of `HLook` and all implementing classes, the term `clipRect` shall be interpreted to mean "clipping rectangle".

A.4.4.17.2 Class description

Only the first paragraph of the "Invocation Mechanism" section shall be considered to be present.

A.4.4.17.3 getPreferredSize

In the description of this method, the text:

- "... then the return value is the size of the largest piece of content..."

shall be considered to be replaced with:

- "...then the return value is a size that is sufficiently large to hold each piece of content..."

The equivalent replacement shall apply for `getMinimumSize()` and `getMaximumSize()`.

In `HLook` and all classes implementing this interface, the following text shall be considered to form part of the method description of getPreferredSize():

- If a default preferred size has been set for this `HVisible` (using `setDefaultSize(Dimension)`) and the default preferred size has an `NO_DEFAULT_WIDTH` then the return value is a `Dimension` with this height (obtained with `getDefaultSize()`) and the preferred width for the content plus any additional dimensions that the `HLook` requires for border decoration etc.
- If a default preferred size has been set for this `HVisible` (using `setDefaultSize(Dimension)`) and the default preferred size has an `NO_DEFAULT_HEIGHT` then the return value is a `Dimension` with this width (obtained with `getDefaultSize()`) and the preferred height for the content plus any additional dimensions that the `HLook` requires for border decoration etc.

A.4.4.17.4 showLook

In `org.havi.ui.HLook.showLook()`, the text:

- The `showLook` method should not modify the `clipRect` of the `Graphics` object that is passed to it.

Shall be clarified as follows:

- The `showLook` method shall not modify the `clipRect` (clipping rectangle) of the `Graphics` object passed to it in a way which includes any area not part of that original `clipRect`. If any modifications are made, the original `clipRect` shall be restored.

A.4.4.18 HMultilineEntry

The following text shall be considered to be added to the class description:

- A call to the inherited method `setDefaultLook(HSinglelineEntry)` shall behave the same as a call to `HSinglelineEntry.setDefaultLook(HSinglelineEntry)`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.19 HMultilineEntryLook

A.4.4.19.1 getCaretCharPositionForLine

The following text:

- If an invalid line is specified an `IllegalArgumentException` is thrown If it cannot be moved the nearest position should be returned.

shall be considered to be replaced by:

- If an invalid line is specified an `IllegalArgumentException` is thrown. If the caret cannot be moved to the same column position on this line, the nearest position should be returned.

A.4.4.19.2 getSoftLineBreakPositions

The following text shall be considered to be added to the returns clause of this method:

- If there is no text content within the `HVisible`, a zero length array shall be returned.

A.4.4.19.3 getVisibleSoftLineBreakPositions

The following text is considered be added to the returns clause of this method:

- If there is no text content within the `HVisible`, a zero length array shall be returned.

A.4.4.20 HNavigable

A.4.4.20.1 Class description

In the following sentence:

- Applications should assume that classes which implement `HNavigable` can only generate events of the type `HFocusEvent` in response to other types of input event.

The word "only" shall be considered to be not present.

A.4.4.20.2 setMove

All occurrences of "then null should be specified" shall be considered to be replaced by "then null shall be specified".

A.4.4.20.3 set FocusTraversal

All occurrences of "then null should be specified" shall be considered to be replaced by "then null shall be specified".

A.4.4.21 HOrientable

A.4.4.21.1 Interface description

The following text:

- All interoperable implementations of the `HOrientable` interface must extend `HComponent`.

Is considered to be replaced with:

- Widgets of HAVi compliant applications implementing the `HOrientable` interface must have `HComponent` in their inheritance tree.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.21.2 getOrientation

The following text:

- The orientation controls how an associated `HLook` lays out the component and affects the visual behaviour of the `HAdjustmentEvent` and `HItemEvent` events. For example, the system might use this information to select appropriate key mappings for these events.

Is considered to be replaced with:

- The orientation controls the layout of the component.

A.4.4.21.3 setOrientation

The following text:

- The orientation controls how the associated `HLook` lays out the component.

Is considered to be replaced with:

- The orientation controls the layout of the component.

A.4.4.21.3.1 Orientation constants

In the description of the orientation constants: `ORIENT_LEFT_TO_RIGHT`, `ORIENT_RIGHT_TO_LEFT`, `ORIENT_TOP_TO_BOTTOM` and `ORIENT_BOTTOM_TO_TOP` the word "shall" is considered to replace "should' in each instance of the phrase "should be rendered".

A.4.4.22 HScene

A.4.4.22.1 addAfter, addBefore

In the returns clause, the text:

- is successfully added;

shall be considered to read:

- is successfully added or was already present.

A.4.4.22.2 getFocusOwner

The following text shall be considered to be added to the end of the method description:

- if and only if this HScene is active.

A.4.4.22.3 Rendering behavior

The following text shall be considered to form an additional paragraph under this heading.

- An application which sets both of setBackgroundMode(NO_BACKGROUND_FILL) and setImageMode(IMAGE_NONE) shall be responsible for ensuring that all pixels in the Hscene are filled with a value which is either opaque or transparent only to video. Such an application cannot make any assumptions about the previous contents of the graphics objects into which it is drawing. For implementations which double-buffer the display of graphics, these existing contents are implementation dependent.

A.4.4.22.4 show

The following text shall be considered to be appended to the description of this method:

- This method does not cause the HScene to request the input focus.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.22.5 setVisible

The following text shall be considered to be present on the end of the method description;

- If this HScene is already visible, then this method brings it to the front. The semantics of show() and setVisible(true) are identical.

A.4.4.23 HScreenConfigurationListener

The parameter "gce" for the report method shall be considered to have the following description.

- gce - The event notifying the listener of the modification.

A.4.4.24 HSceneFactory

A.4.4.24.1 getDefaultHScene

The text:

- identical to calling org.havi.ui.HScene.getDefaultHscene(

shall be considered to read:

- identical to calling org.havi.ui.HSceneFactory.getDefaultHscene(

A.4.4.24.2 Class Description

The following text:

- Calling resizeScene for an HScene shall apply the same policies as described above for newly created HScenes when deciding whether the resizing operation requested is possible.

Shall be considered to read:

- Calling resizeScene for an HScene shall apply the same policies as described above for newly created HScenes when deciding whether the method call is possible.

A.4.4.25 HSelectionInputPreferred

A.4.4.25.1 Interface description

The following text is considered to be added to the interface description:

- The system must provide a means of generating `HItemEvent` events as necessary. For platforms with a restricted number of physical keys this may involve a "virtual keyboard" or similar mechanism.The system might use the information returned by the method `getOrientation()` of the super interface to select appropriate key mappings for this event. The mechanisms to generate this event shall not be effective while the component is disabled (see `HComponent.setEnabled()`).

Also, the text:

- All interoperable implementations of the `HSelectionInputPreferred` interface must extend `HComponent`.

Is considered to be replaced with:

- Widgets of HAVi compliant applications implementing the `HSelectionInputPreferred` interface must have `HComponent` in their inheritance tree.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Also, in the text:

- Note that the `java.awt.Component` method `isFocusTraversable` should always return true for a `java.awt.Component` implementing this interface.

The word "should" is considered to be replaced with "shall".

A.4.4.25.2 getSelectionMode

The description of this method is considered to be extended by the following text:

- Note that these events are ignored, if the component is disabled. See also: `HComponent.setEnabled()`.

A.4.4.25.3 processHItemEvent

The description of this method is considered to be extended by the following text:

- Widgets implementing this interface shall ignore HItemEvents, while the component is disabled. See also: `HComponent.setEnabled()`.

A.4.4.25.4 setSelectionMode

The description of this method is considered to be extended by the following text:

- Calls to this method shall be ignored, if the component is disabled. See also: `HComponent.setEnabled()`.

A.4.4.26 HSingleLineEntry

A.4.4.26.1 Class description

The term:

- `TEXT_START_CHANGE` event

is considered to be replaced with:

- an `HTextEvent` event with an id of `TEXT_START_CHANGE`.

A.4.4.26.2 Default parameter values not exposed in the constructors

In the row "Password protection (the echo character)", in the "default value" column, replace

- Entry is "clear", i.e. not password protected.

with:

- Zero (ASCII NUL), i.e. not password protected.

A.4.4.26.3 setType(int)

The following text:

- Set the type of permitted keyboard entry.

  Parameters:

  - type - one of INPUT_ANY, INPUT_ALPHA, INPUT_NUMERIC or INPUT_CUSTOMIZED

Shall be considered to be replaced with:

- Set to indicate to the system which input keys are required by this component. The input type constants can be added to define the union of the character sets corresponding to the respective constants.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Parameters:

- type - sum of one or several of INPUT_ANY, INPUT_NUMERIC, INPUT_ALPHA, or INPUT_CUSTOMIZED.

A.4.4.26.4 getValidInput

The following text:

- Retrieve the customized input character range. The return value of this method should reflect the range of input keys which the component wishes to see, should getType return a value with the INPUT_CUSTOMIZED bit set. This method may return null if customized input is not requested.

Shall be considered to be replaced with:

- Retrieve the customized input character range. If getType returns a value with the INPUT_CUSTOMIZED bit set then this method shall return an array containing the range of customized input keys. If the range of customized input keys has not been set then this method shall return a zero length char array. This method shall return null if getType returns a value without the INPUT_CUSTOMIZED bit set.

A.4.4.27 HStaticAnimation

The following text is considered to be added to the description of this class:

- Calling `setVisible(false)` on `HStaticAnimation` shall not automatically stop the animation hence applications are not required to call the `play()` method again when the animation again becomes visible. It is implementation dependent whether the animation continues from the last visible position or from where it would be if it kept running.

A.4.4.28 HStaticRange

A.4.4.28.1 Fields

A.4.4.28.1.1 SCROLLBAR_BEHAVIOR

The description of this field is considered to be replaced by the following text:

- The `HStaticRange` shall behave as a scrollbar, i.e. the allowable values that may be set / returned for the `HStaticRange` shall be affected by the "thumb" offsets, and hence its value shall be able to vary between `[minimum + minThumbOffset, maximum - maxThumbOffset]`.

A.4.4.28.1.2 SLIDER_BEHAVIOR

The description of this field is considered to be replaced by the following text:

- The `HStaticRange` shall behave as a slider, i.e. the allowable values that may be set / returned for the `HStaticRange` shall not be affected by the "thumb" offsets, and hence its value shall be able to vary between `[minimum, maximum]`.

A.4.4.28.2 getOrientation

The following text:

- The orientation controls how an associated `HLook` lays out the component and affects the visual behaviour of the `HAdjustmentEvent` and `HItemEvent` events. For example, the system might use this information to select appropriate key mappings for these events.

Is considered to be replaced by:

- The orientation controls how the associated `HLook` lays out the component.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.28.3 setBehavior

The following text is considered to be appended to the method description:

- If the new behaviour is `SCROLLBAR_BEHAVIOR` and the control's settings for range and thumb offsets are illegal, i.e. `minimum` + `thumbMinOffset` is equal or greater than `maximum` - `thumbMaxOffset`, then an `IllegalArgumentException` shall be thrown. If the control's value is not valid for the offsets, then the value shall be changed to the closest valid value.

A.4.4.28.4 setRange

The following text is considered to be appended to the method description:

- If the maximum is greater than the minimum and the value of the control is outside the new range (subject to the control's current behaviour), then the value is changed to the closest valid value.

A.4.4.28.5 setThumbOffsets

The following text is considered to be appended to the method description:

- If this control's behaviour is `SCROLLBAR_BEHAVIOR`, then the following rules apply: If the thumb offsets are illegal, i.e. `minimum` + `thumbMinOffset` is equal or greater than maximum - `thumbMaxOffset`, then an `IllegalArgumentException` shall be thrown. If the control's value is not valid for the specified offsets, then the value shall be changed to the closest valid value.

A.4.4.28.6 setValue

The method's parameter description is considered to be replaced by:

- value - the value for this HStaticRange.

Also, the following text is considered to be appended to the method description:

- If the specified value is not valid, then the method shall round it to the closest valid value. An application can retrieve the corrected value by means of method `getValue()`.

A.4.4.29 HVideoDevice

A.4.4.29.1 getBestConfiguration

The following text is considered to be added to the end of the second paragraph of the descriptions of both methods of this name.

- If there are such equally best configurations, the one which is returned by this method is an implementation dependent selection from among those which are equally best.

A.4.4.29.2 setVideoConfiguration

In the `setVideoConfiguration` method, the sentence below:

- Applications can prevent or limit changes to configurations of other, not intended, devices by using constants ZERO_GRAPHICS_IMPACT and ZERO_VIDEO_IMPACT in their configuration templates.

shall be considered to be extended as follows:

- Applications can prevent or limit changes to configurations of other, not intended, devices by using constants ZERO_GRAPHICS_IMPACT, ZERO_VIDEO_IMPACT and ZERO_BACKGROUND_IMPACT in their configuration templates.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The following additional text is considered to be present.

NOTE: If a configuration which includes ZERO_GRAPHICS_IMPACT or ZERO_BACKGROUND_IMPACT and this would require changes to already running devices then this will not be possible to apply successfully and hence this method will return False.

A.4.4.29.3 getVideoController

The following text:

- HPermissionDeniedException - (HPermissionDeniedException) if the application does not currently have the right to get the VideoPlayer object.

shall be considered to be replaced by:

- HPermissionDeniedException - (HPermissionDeniedException) this exception shall never be thrown.

A.4.4.29.4 setVideoConfiguration

The following shall be added as the first item in the bulleted list:

- If an application tries to select a configuration which is not valid for that device at that time or when the device is in a particular mode then an HConfigurationException shall be thrown.

A.4.4.29.5 getConfigurations

The following shall be added to the end of the method description.

- The set of configurations returned may include ones which are only valid for the device at particular times or when the device is in a particular mode.

A.4.4.29.6 getVideoSource

The following text:

- HPermissionDeniedException - if the application does not currently have the right to get the `VideoSource` object.

Shall be considered to be replaced by:

- HPermissionDeniedException - this exception shall never be thrown.

A.4.4.29.7 Constructor

In the constructor of this class, the following sentence:

- It is not intended that applications should directly construct `HVideoDevice` objects.

Shall be considered to be replaced with the following:

- An interoperable application shall not subclass the `HVideoConfiguration` class.

A.4.4.30 HVisible

A.4.4.30.1 Class description

In the class description of `HVisible`, the sentence:

- Most content is stored by reference, but text content is copied as `java.lang.String` objects are immutable.

Shall be considered to not be present.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

In the table "Default parameter values not exposed in the constructors", row "getDefaultSize", column "Default value" the following text:

- The default preferred size not set (i.e. null) unless specified by width and height parameters.

Shall be considered to be replaced with:

- The default preferred size not set (i.e. NO_DEFAULT_SIZE) unless specified by width and height parameters.

A.4.4.30.2 Constructors

The following text shall be considered to form part of those constructor descriptions which have an `HLook` as a parameter:

- Applications shall not use `HLooks` with this constructor unless those `HLooks` are specified as working with `HVisible`. If an `HLook` is used which is specified as only working with specific sub-classes of `HVisible` then the failure mode is implementation dependent.

A.4.4.30.3 setDefaultSize

The following text:

- If this parameter or specifies a size smaller than an implementation-defined minimum size a `java.lang.IllegalArgumentException` will be thrown.

Shall be considered to be replaced by:

- If this parameter specifies a size smaller than an implementation-defined minimum size, the preferred size of this component shall be set to that implementation-defined minimum size.

Also, the following text:

- If the parent Container into which the `HVisible` is placed has no layout manager this method has no effect.

Is considered to be removed.

A.4.4.30.4 setLookData

The method description shall be considered to have the following text added.

- If for this specified key a data object has been set, the old data object shall be replaced with the new one.

A.4.4.30.5 setResizeMode

The following text added shall be considered to be replaced:

- Set the scaling mode for scaling any state-based content rendered by an associated HLook. If content is not used in the rendering of this HVisible calls to this method shall change the current alignment mode, but this will not affect the rendered representation.

by:

- Set the resize mode of this HVisible. If the associated HLook does not render content or if scaling is not supported, changing the mode may have no visible effect.

A.4.4.30.6 setTextContent

In the description of this method, the following sentence is considered to not be present:

- Note that unlike `setGraphicContent`, `setAnimateContent` and `setContent`, the content is copied as it is not possible to store a reference to a `java.lang.String`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.30.7 NO_DEFAULT_SIZE

The following text shall be considered to be appended to the end of the field description.

- The contents of the Dimension object cannot be relied upon. Comparisons must always be done using object identity, i.e. using the "==" operator.

A.4.4.31 HVideoConfigTemplate

A.4.4.31.1 setPreference

The paragraph:

- An application which wishes to remove a preference from an existing template (e.g. one generated by the platform) may call this method with null for the object parameter.

Shall be considered to be extended with the following sentence:

- Specifying null as the object parameter shall have no effect if the preference is not in the template.

A.4.4.32 HVideoComponent

A.4.4.32.1 HAVi Specification

In section 8.3.3.6 "Integrating HAVi Video Support into Platforms" the following two sentences shall be removed:

- "The class HVideoComponent is intended to be returned by a platform specific controller for video. In platforms based on the Java Media Framework, the Player.getVisualComponent method shall return objects of this class".

Note that the `HVideoComponent` class must be present, and MHP terminals may choose to use it or not. Interoperable applications should not use `HVideoComponent`.

A.4.4.32.2 removeOnScreenLocationModifiedListener

In this method, the parameters clause shall be changed from:

- slml - listener to be notified when the on-screen location of the component is modified.

to:

- slml - listener to be removed

A.4.4.33 HBackgroundConfiguration

A.4.4.33.1 setColor

The "throws" clause for `HPermissionDeniedException` shall be considered to read as follows

- If the application has not currently reserved the HBackgroundDevice associated with this configuration or this configuration is not the current configuration of that HBackgroundDevice.

A.4.4.33.2 getConfigTemplate

The following text shall be considered to be added to the description of this method:

- Preferences that are not filled in by the platform will return DONT_CARE priority.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.33.3 Constructor

In the description of the constructor, the following sentence:

- It is not intended that applications should directly construct `HBackgroundConfiguration` objects.

shall be considered to read;

- An interoperable application shall not subclass the `HBackgroundConfiguration` class.

A.4.4.34 HScreenDevice

A.4.4.34.1 reserveDevice

The following paragraph:

- Once the right to control this device has been granted and not removed in the intervening period further calls to this method shall have no effect and return true.

Shall be considered to be replaced by:

- Once the right to control this device has been granted and not removed in the intervening period further calls to this method re-using the current resource client shall have no effect and return true.

A.4.4.35 HImageMatte

A.4.4.35.1 setMatteData

The method `setMatteData` shall be considered to have the following text added:

- Note that if the size of the image is smaller than the size of the component to which the matte is applied, the empty space behaves as if it were an opaque flat matte of value 1.0. By default images are aligned at the top left corner of the component. This can be changed with the setOffset method.

A.4.4.36 HVersion

A.4.4.36.1 General

The following text:

- Note that it is a valid implementation to return empty strings for the implementation, vendor and name strings.

Shall be considered to be replaced with:

- Note that it is a valid implementation to return empty strings for HAVI_IMPLEMENTATION_NAME, HAVI_IMPLEMENTATION_VENDOR and HAVI_IMPLEMENTATION_VERSION strings.

A.4.4.36.2 MHP Specific Clarification

In MHP, a call to `getProperty()` when referencing the constants listed in column Constant in the table below shall return a string as listed in column Value.

Table 67: getProperty

Constant	Value
HAVI_SPECIFICATION_VENDOR	"DVB"
HAVI_SPECIFICATION_NAME	"MHP"
HAVI_SPECIFICATION_VERSION	"<version>"

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

<version>

This string shall represent the MHP version to which this HAVi implementation is conformant. The encoding shall be, in order, the major, minor and micro values presented as the textual representation of decimal integers with no leading zeros and separated by a single "." character, e.g. "1.0.3".

A.4.4.36.3 Field descriptions

- Consider all occurrences of: java.lang.System.getProperty(havi.specification.version) to be replaced with: java.lang.System.getProperty(HVersion.HAVI_SPECIFICATION_VERSION); and
- all occurrences of: java.lang.System.getProperty(havi.specification.vendor) to be replaced with: java.lang.System.getProperty(HVersion.HAVI_SPECIFICATION_VENDOR); and
- all occurrences of: java.lang.System.getProperty(havi.specification.name to be replaced with: java.lang.System.getProperty(HVersion.HAVI_SPECIFICATION_NAME);
- all occurrences of: java.lang.System.getProperty(havi.implementation.version) to be replaced with: java.lang.System.getProperty(HVersion.HAVI_IMPLEMENTATION_VERSION); and
- all occurrences of: java.lang.System.getProperty(havi.implementation.vendor) to be replaced with: java.lang.System.getProperty(HVersion.HAVI_IMPLEMENTATION_VENDOR); and
- all occurrences of: java.lang.System.getProperty(havi.implementation.name) to be replaced with: java.lang.System.getProperty(HVersion.HAVI_IMPLEMENTATION_NAME).

A.4.4.37 HKeyboardInputPreferred

A.4.4.37.1 INPUT_NUMERIC

The following text:

- This constant indicates that the component only requires alphanumeric input, as determined by the java.lang.Character isDigit method.

shall be considered to be replaced by:

- This constant indicates that the component requires numeric input, as determined by the java.lang.Character isDigit method.

A.4.4.37.2 INPUT_ALPHA

The following text:

- This constant indicates that the component only requires alphanumeric input, as determined by the java.lang.Character isLetter method.

Shall be considered to be replaced by:

- This constant indicates that the component requires alphabetic input, as determined by the java.lang.Character isLetter method.

A.4.4.37.3 getValidInput

The following text:

- Retrieve the customized input character range. The return value of this method should reflect the range of input keys which the component wishes to see, should getType return a value with the INPUT_CUSTOMIZED bit set. This method may return null if customized input is not requested.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Shall be considered to be replaced with:

- Retrieve the customized input character range. If getType returns a value with the INPUT_CUSTOMIZED bit set then this method shall return an array containing the range of customized input keys. If the range of customized input keys has not been set then this method shall return a zero length char array. This method shall return null if getType returns a value without the INPUT_CUSTOMIZED bit set.

A.4.4.38 HScreenConfigTemplate

A.4.4.38.1 Class description

The following text shall be considered to be added to the end of the class description:

- Several preferences in this class are required to be not filled in by the platform in templates generated by the platform. This shall mean:
  - the object for this preference (if there is one) is set to null;
  - the priority for this preference is set to DONT_CARE.

A.4.4.38.2 VIDEO_GRAPHICS_PIXEL_ALIGNED

In the description of this field, the following text shall be considered to be absent:

- Where a video device is moving the video relative to the screen in real time (e.g. implementing pan and scan), graphics configurations shall only support this feature where the implementation of the graphics device can track the position changes in the video device automatically.

A.4.4.38.3 DISABLED

Add the following to the class description of HScreenConfigTemplate

```
*
* A special template indicating that an HScreenDevice is disabled.
* Disabled HScreenDevice instances are not being composited as part of
* the video output signal of the HScreen.
* The preferences, properties and attributes of this HScreenConfigTemplate
* are implementation dependent.
*
public static HScreenConfigTemplate DISABLED;
```

A.4.4.38.4 SCREEN_ASPECT_RATIO

The following additional field shall be considered to be present.

```
/**
 * A value for use in the preference field of the setPreference,
 * getPreferenceObject and getPreferencePriority methods in the
 * HScreenConfigTemplate that indicates that the device configuration
 * supports the screen aspect ratio, as specified in a Dimension object
 * which indicates the (relative) x, y pixel aspect ratio. Typical
 * values are 4:3 and 16:9. <p>
 *
 * Instances of HScreenConfigTemplate generated by the platform and
 * returned to applications (e.g. from getConfigTemplate shall have this
 * preference set to a platform specific value with the REQUIRED
 * priority.
 */
public static final int SCREEN_ASPECT_RATIO=16;
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.39 HGraphicsConfiguration

A.4.4.39.1 getConfigTemplate

The following text shall be considered to be added to the description of this method:

- Preferences that are not filled in by the platform will return DONT_CARE priority.

A.4.4.39.2 Constructor

In the constructor of this class, the following sentence:

- It is not intended that applications should directly construct `HGraphicsConfiguration` objects.

Shall be considered to be replaced with the following:

- An interoperable application shall not subclass the `HGraphicsConfiguration` class.

A.4.4.40 HVideoConfiguration

A.4.4.40.1 getConfigTemplate

The following text shall be considered to be added to the description of this method:

- Preferences that are not filled in by the platform will return DONT_CARE priority.

A.4.4.40.2 Constructor

In the constructor of this class, the following sentence:

- It is not intended that applications should directly construct `HVideoConfiguration` objects.

shall be considered to be replaced with the following:

- An interoperable application shall not subclass the `HVideoConfiguration` class.

A.4.4.41 HToggleButton

A.4.4.41.1 Default parameter values exposed in the constructors

The following descriptions will be used to replace the ones already in HToggleButton's "Default parameter values exposed in the constructors" table:

- imageFocused - The image to be used as the content for the HState.FOCUSED_STATE and HState.DISABLED_FOCUSED_STATE states of this component.
- imageActioned - The image to be used as the content for the HState.ACTIONED_FOCUSED_STATE and HState.DISABLED_ACTIONED_FOCUSED_STATE states of this component.
- imageNormalActioned - The image to be used as the content for the HState.ACTIONED_STATE and HState.DISABLED_ACTIONED_STATE states of this component.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.42 HGraphicButton

A.4.4.42.1 Default parameter values exposed in the constructors

The following descriptions will be used to replace the ones already in HGraphicButton's "Default parameter values exposed in the constructors" table:

- imageFocused - The image to be used as the content for the HState.FOCUSED_STATE and HState.DISABLED_FOCUSED_STATE states of this component.
- imageActioned - The image to be used as the content for the HState.ACTIONED_STATE, HState.ACTIONED_FOCUSED_STATE states of this component.

A.4.4.43 HTextButton

A.4.4.43.1 Default parameter values exposed in the constructors

The rows for textFocus and textAction in HTextButton's "Default parameter values exposed in the constructors" table shall be considered to be removed.

A.4.4.44 HLook and classes implementing HLook

In the getMinimumSize method of HLook and all these classes, the following paragraph shall be considered to be replaced:

- If the HLook supports the scaling of its content (e.g. an HGraphicLook) and content is set then the return value is a size sufficiently large to hold the minimum size of each piece of content plus any additional dimensions that the HLook requires for border decoration etc.

with:

- If the HLook supports the scaling of its content (e.g. an HGraphicLook) and scaling is requested and content is set then the return value is a size containing the width of the narrowest content and the height of the shortest content plus any additional dimensions that the HLook requires for border decoration etc.

A.4.4.45 HTextLook

A.4.4.45.1 General

In getMaximumSize(), getMinimumSize() and getPreferredSize(), the following sentence:

- If the HDefaultTextLayoutManager returns a zero size, then proceed with the following steps.

shall be considered to be replaced with the following:

- If HVisible.getTextLayoutManager() does not return an instance of HDefaultTextLayoutManager, or HDefaultTextLayoutManager returns a zero size, then proceed with the following steps as if content for this HVisible had not been set.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.46 HExtendedLook

org.havi.ui

HExtendedLook

Syntax

```
public interface HExtendedLook extends HLook
```

All Superinterfaces:

java.lang.Cloneable, `HLook`

Description

The HExtendedLook interface is derived from the `HLook` interface and abstracts out the drawing of the look into separate background, border and visible data components. The interface allows the programmer to derive new looks from the default looks and have control over which component parts are drawn. This aids interoperability between platforms since no two manufacturer's `HLook` implementation look the same.

See Also:

setLook(HLook), setLookData(Object, Object), paint(Graphics), setBackgroundMode(int), setHorizontalAlignment(int), setVerticalAlignment(int), setResizeMode(int), HTextLook, HGraphicLook, HAnimateLook, HRangeLook, HSinglelineEntryLook, HMultilineEntryLook

Methods

fillBackground(Graphics, HVisible, int)

```
public void fillBackground(java.awt.Graphics g, HVisible visible, int state)
```

The fillBackground(Graphics, HVisible, int) method paints the component with its current background `Color` according to the `setBackgroundMode(int)` method of `HVisible`.

This method is only called from `showLook` within this `HExtendedLook` implementation. It's not the intention to call this method directly from outside of the `HExtendedLook`.

Regardless of the background mode, it is an implementation option for this method to render added decorations which may affect the look's transparency. This method shall not be used to render any HVisible related data or content associated with the HVisible. It is purely for background and decoration only.

The `fillBackground` method should not modify the clipRect (clipping rectangle) of the `Graphics` object that is passed to it in a way which includes any area not part of that original clipRect. If any modifications are made, the original clipRect shall be restored.

Parameters:

`g` - the graphics context.

`visible` - the visible.

`state` - the state parameter indicates the state of the visible.

See Also:

`isOpaque(HVisible)`

renderBorders(Graphics, HVisible, int)

```
public void renderBorders(java.awt.Graphics g, HVisible visible, int state)
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The renderBorders(Graphics, HVisible, int) method paints any implementation specific borders according to the `setBordersEnabled(boolean)` method of `HVisible`. If borders are drawn, the border decoration shall be rendered within the border area as returned by `getInsets`. The behavior of this method, when a subclass overrides the method `getInsets` is undefined, except that it will never draw outside the border area as returned by `getInsets`.

This method is only called from `showLook` within this `HExtendedLook` implementation. It's not the intention to call this method directly from outside of the `HExtendedLook`.

The renderBorders(Graphics, HVisible, int) method should not modify the clipRect (clipping rectangle) of the `Graphics` object that is passed to it in a way which includes any area not part of that original clipRect. If any modifications are made, the original clipRect shall be restored.

Parameters:

`g` - the graphics context.

`visible` - the visible.

`state` - the state parameter indicates the state of the visible.

renderVisible(Graphics, HVisible, int)

```
public void renderVisible(java.awt.Graphics g, HVisible visible, int state)
```

The renderVisible(Graphics, HVisible, int) method paints any content or other data associated with the look's HVisible. This method shall not be used to render a background nor any other decoration. Its purpose is purely to render content or other value data stored on the HVisible. This may be set via HVisible methods such as `setTextContent` and `setGraphicContent`. Rendering shall take place within the bounds returned by the `getInsets` method.

This method is only called from `showLook` within this `HExtendedLook` implementation. It's not the intention to call this method directly from outside of the `HExtendedLook`.

For looks which draw content (e.g. `HTextLook`, `HGraphicLook` and `HAnimateLook`), if no content is associated with the component, this method does nothing.

The renderVisible(Graphics, HVisible, int) method should not modify the clipRect (clipping rectangle) of the `Graphics` object that is passed to it in a way which includes any area not part of that original clipRect. If any modifications are made, the original clipRect shall be restored.

Parameters:

`g` - the graphics context.

`visible` - the visible.

`state` - the state parameter indicates the state of the visible.

showLook(Graphics, HVisible, int)

```
public void showLook(java.awt.Graphics g, HVisible visible, int state)
```

The showLook(Graphics, HVisible, int) method is the entry point for repainting the entire `HVisible` component. This method delegates the responsibility of drawing the component background, borders and any `HVisible` related data or content to the `fillBackground`, `renderVisible` and `renderBorders` methods respectively, subject to the clipping rectangle of the Graphics object passed to it. This method shall call the methods `fillBackground`, `renderVisible`, and `renderBorders` in that order and shall not do any rendering itself.

The showLook(Graphics, HVisible, int) method should not modify the clipRect (clipping rectangle) of the `Graphics` object that is passed to it in a way which includes any area not part of that original clipRect. If any modifications are made, the original clipRect shall be restored.

Any resources explicitly associated with an HExtendedLook should be loaded by the HExtendedLook during its creation, etc. Note that the "standard" looks don't load content by default.

This method is called from the `paint(Graphics)` method of `HVisible` and must never be called from elsewhere. Components wishing to redraw themselves should call their repaint method in the usual way.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Overrides:

showLook(Graphics, HVisible, int) in interface HExtendedLook

Parameters:

`g` - the graphics context.

`visible` - the visible.

`state` - the state parameter indicates the state of the visible, allowing the look to render the appropriate content for that state. Note that some components (e.g. HStaticRange, HRange, HRangeValue) do not use state-based content.

A.4.4.47 HAnimateLook, HGraphicLook, HListGroupLook, HMultilineEntryLook, HRangeLook, HSinglelineEntryLook and HTextLook

The following changes shall be considered to apply to all these implementing looks.

a) The method descriptions in HExtendedLook for fillBackground(), renderBorders() and renderVisible() shall be considered to form part of all the above looks.

b) The description of showLook() in each of the above looks shall be considered to be replaced with the one in HExtendedLook.

c) HAnimateLook, HGraphicLook, HSinglelineEntryLook and HTextLook shall be considered to implement HExtendedLook instead of HLook.

d) HListGroupLook and HRangeLook shall be considered to implement HExtendedLook additionally to HAdjustableLook. The method showLook() shall be implemented as specified by HExtendedLook.

A.4.4.48 HSwitchable

The following sentence:

- Actioned states (i.e. those with the ACTIONED_STATE_BIT bit set may persist after any registered HActionListener listeners have been called, until a further HActionEvent is received.

Shall be considered to be modified as follows:

- Actioned states (i.e. those with the ACTIONED_STATE_BIT bit set) may persist after any registered HActionListener listeners have been called, until a further HActionEvent is received.

A.4.4.49 HStillImageBackgroundConfiguration

A.4.4.49.1 Constructor

In the constructor of this class, the following sentence:

- It is not intended that applications should directly construct `HStillImageBackgroundConfiguration` objects.

Shall be considered to be replaced with the following:

- An interoperable application shall not subclass the `HStillImageBackgroundConfiguration` class.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.50 HTextValue

A.4.4.50.1 Class description

In the description of this interface, the following text:

- HAVi text events are discussed in detail in the HKeyboardInputPreferred interface description.

shall be considered to be replaced by:

- HAVi text events are discussed in detail in the HTextEvent description.

A.4.4.51 HDefaultTextLayoutManager

A.4.4.51.1 getMinimumSize

The following parameters and returns clause shall be considered to be present:

Parameters:

- hvisible - a component within which text will be rendered.

Returns:

- the minimum size of text rendered in that component.

A.4.4.51.2 getMaximumSize

The following parameters and returns clause shall be considered to be present.

Parameters:

- hvisible - a component within which text will be rendered.

Returns:

- the maximum size of text rendered in that component.

A.4.4.51.3 getPreferredSize

The following parameters and returns clause shall be considered to be present.

Parameters:

- hvisible - a component within which text will be rendered.

Returns:

- the preferred size for text rendered in that component.

A.4.4.52 Hscreen

A.4.4.52.1 getHGraphicsDevices, getHVideoDevices, getHBackgroundDevices

Add the following to the method body of each of the following methods - HScreen.getGraphicsDevices, HScreen.getVideoDevices, HScreen.getBackgroundDevices:

```
*
* The devices returned shall be in z-order with each device being in
* front of those devices with a higher index in the array of results.
*
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.4.52.2 getCoherentScreenConfigurations

Add the following to the method description of HScreen.getCoherentScreenConfigurations(..)

```
*
* Both the input to this method and the output from this method shall be
* sorted in z-order (from front to back) within each type of HscreenConfiguration.
* The ordering of different sub-classes of HScreenConfigTemplate shall be ignored.
* Where more than one template of the same type is
* provided, the returned configurations shall be in the same z-order as
* the templates in the input. e.g. if the template at index 0 in the input
* requests a resolution of 960x540 and the template at index 1 in the
* input requests a resolution of 720x576 then the configuration at index 0
* in the output shall have a resolution of 960x540 and the configuration
* at index 1 in the output shall have a resolution of 720x576. If the
* z-order requirement cannot be met, this function shall return null.
* The special HScreenConfigTemplate DISABLED should not normally be requested
* in this method since more capable MHP terminals may be able to support
* more than what is requested.
*
```

A.4.4.52.3 setCoherentScreenConfigurations

In the method, HScreen.setCoherentScreenConfigurations():

Add the following to the class description:

- The HScreenConfigurations shall be in z-order as defined for HScreen.getCoherentScreenConfigurations.

Change the @throws clause for HConfigurationException to read as follows:

```
*
* @throws HConfigurationException - if the specified
* HScreenConfiguration[] array is not valid for any of the devices
* or if the z-order of the configurations in the array does not match
* the z-order of the devices.
*
```

A.4.4.52.4 Additional methods

The following additional methods shall be considered to be present:

```
/**
 * Return the aspect ratio of the video output signal represented
 * by this HScreen.
 * @return a Dimension object specifying the aspect ratio of the video
 * output signal
 */
public Dimension getScreenAspectRatio()

/**
 * Return the aspect ratio of the the physical display that is
 * connected to this HScreen as far as that is known. The extent to
 * which this is known depends on the connector and protocol used
 * to connect the display. Some connectors may support a protocol
 * signalling the display aspect ratio. Otherwise the display aspect
 * ratio may be a user setting entered as part of the configuration
 * process of the MHP terminal.
 *
 * @return a Dimension object specifying the aspect ratio of the display
 * @see VideoFormatControl#getDisplayAspectRatio
 */
public Dimension getDisplayAspectRatio()
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.5 org.havi.ui.event

A.4.5.1 HActionEvent

A.4.5.1.1 getModifiers

The text:

- Modifiers are not used with the HAVi platform. Interoperable HAVi applications shall not use the return value of this method.

Shall be considered to be replaced by:

- Modifiers are not used for `HActionEvents` with the HAVi platform. Interoperable HAVi applications shall not use the return value of this method.

A.4.5.2 HItemEvent

A.4.5.2.1 Constructor

The following text:

- item - The item which caused the change, or null if this information is not available. This information shall be provided if the event id is one of ITEM_SELECTED, ITEM_CLEARED, ITEM_SET_CURRENT, ITEM_SET_NEXT or ITEM_SET_PREVIOUS.

Shall be considered to be replaced with the following:

- item - The item which caused the change, or null if this information is not available.

  If the event is sent to listeners, this information shall be provided if the event id is one of ITEM_SELECTED, ITEM_CLEARED, ITEM_SET_NEXT or ITEM_SET_PREVIOUS".

A.4.5.2.2 ITEM_SET_CURRENT

The following paragraph:

- An item event with this id is sent from the component whenever the current item of an HItemValue component changes.

Shall be considered to be replaced with:

- An item event with this id is sent to or from the component whenever the current item of an HItemValue component changes.

A.4.5.3 HKeyEvent

The following paragraph shall not apply in the present document:

- Note that the HAVi system should only generate KEY_PRESSED events. Neither KEY_TYPED nor KEY_RELEASED events should be generated. Furthermore, the system should collapse combined events. For example, a usual Java Virtual Machine generates for the letter A three events: KEY_PRESSED for modifier key Shift, KEY_PRESSED for letter "A" and KEY_TYPED for "A". This should be collapsed into one single KEY_PRESSED event with the letter "A" and the Shift modifier set. This is to simplify the key event handling of applications.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.5.3.1 Constructors

The description of the "id" parameter shall be considered to have the following text added:

- This is the value that will be returned by the event object's getID method.

A.4.5.4 HRcEvent

The following changes shall be considered to apply in this class:

- Replace all occurrences of "action id" in the descriptions of the VK constants by "key code".
- Replace all occurrences of "event ids" in the descriptions of RC_FIRST and RC_LAST by "key codes".Tag both constants as being deprecated. Add for RC_FIRST a specific deprecated text:

  * @deprecated The value of this field is useless, since it mixes event ids and key codes. It does not reflect any of the remote control key codes listed in this class.
- Remove the following sentence from the class description: "Note that it is an implementation constraint that the HrcEvent event range should not intersect with the Java AWT key event range".
- Add to the parameter descriptions of "id" for both constructors: "in the range KEY_FIRST to KEY_LAST".
- In the definition of the field, VK_COLORED_KEY_0, the paragraph starting with "Up to six colored soft keys can be included on a remote control." shall be considered to be extended with the following; "In markets where text is written from right to left, these keys may be oriented from right to left in ascending order".

A.4.5.5 HRcCapabilities

A.4.5.5.1 getRepresentation

In the method `getRepresentation(int aCode)`, all references to the parameter `id` shall be considered to be replaced with `keyCode`.

A.4.5.5.2 Constructor

The following sentence:

- It is not intended that applications should directly construct `HRcCapabilities` objects.

Shall be considered to be replaced by:

- An interoperable application shall not subclass the `HRcCapabilities` class.

A.4.5.6 HEventGroup

A.4.5.6.1 getKeyEvents

This class shall be considered to be extended with the following method:

```
/**
 * Return the key codes contained in this event group
 */
public int[] getKeyEvents() {}
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

A.4.5.7 HKeyCapabilities

A.4.5.7.1 Constructor

The following sentence:

- It is not intended that applications should directly construct `HKeyCapabilities` objects.

Shall be considered to be replaced by:

- An interoperable application shall not subclass the `HKeyCapabilities` class.

A.4.5.8 HEventRepresentation

A.4.5.8.1 Constructor

The following sentence:

- It is not intended that applications should directly construct `HEventRepresentation` objects.

Shall be considered to be replaced by:

- An interoperable application shall not subclass the `HEventRepresentation` class.

A.4.6 References to ISO/IEC10646-1:1993

All references to ISO/IEC10646-1:1993 are considered to be non-specific references to ISO/IEC 10646-1 [15]. Also, the term "support for Unicode ranges" is considered to be "support for character ranges as specified by ISO/IEC 10646-1 [15]".

A.4.7 Chapter 8

A.4.7.1 Section 8.2.3.3.2 "Compatibility with Existing java.awt Methods"

In this section, the following sentence:

- The `java.awt.Toolkit.getScreenSize` method shall be equivalent to the pixel resolution of the current configuration of the default screen device returned by HScreen.getDefaultGraphicsDevice.

Shall be considered to read:

- The `java.awt.Toolkit.getScreenSize` method shall be equivalent to the pixel resolution of the current configuration of the default screen device returned by HScreen.getDefaultHGraphicsDevice.

A.5 ISO/IEC 13818-6

With reference to ISO/IEC 13818-6 [38].

A.5.1 Reconstruction of NPT

In equation 8-3, SCR_Reference is considered to be STC_Reference.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Annex B (normative): Broadcast filesystem and trigger transport

B.0 General

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

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

B.1 Service domain

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

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

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

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

B.2 Filesystem requirements

B.2.1 Static requirements

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

Table 68: Filesystem signalling requirements

Area	Minimum requirement
Characters Allowed in File and Directory Names	The ASCII character "a.."z", "A.."Z", "0.."9", "$", "-", and "_". After the first character of a file name, "." and " " (the space character) is also permitted
Maximum length of file name	200 characters
Number of entries per directory	500
Maximum Directory Nesting	20 levels
Maximum File Length	2^{27} - 1 bytes
Caching	See clause B.2.1.1, "Caching behaviour"

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

B.2.1.1 Caching behaviour

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

This signalling shall contain sufficient information to derive the following.

Table 69: Caching behaviour signalling requirements

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

priority_value: indicates the caching priority for the objects within this module. A higher value indicates more importance for caching.

transparency_level: Transparency level that shall be used by the GEM terminal if it caches objects contained in this module. The possible values are listed in table 70. The semantics of the policies are defined in clause B.5.2, "Transparency levels of caching".

Table 70: Transparency level values

Value	Description
0	Reserved
1	Transparent caching
2	Semi-transparent caching
3	Static caching.
4 to 255	Reserved for future use

When not otherwise specified, the default values that shall be assumed are:

- priority_value: 128.
- transparency_level: 1 (transparent caching).

This clause describes the default semantics of these two fields. Additional semantics are defined in clause B.5. GEM terminals that support caching shall comply with these semantics.

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

B.2.2 Filesystem updates

For broadcast targets, it shall be possible to signal a new version of a file or directory.

B.2.3 Content type descriptor

Zero or one content type descriptors can be present in the filesystem. Where more than one content type descriptor is used they shall express the same content format. Also, the content type (if any) signalled in the directory binding shall be identical to that signalled in the bound file's header. This optional descriptor identifies the media type of the file.

This content type signalling only applies to objects of type file and is not appropriate for other object types.

If this descriptor is absent or not sufficient to categorize the content type then the extension portion of the file name shall be used be used to provide the media type mapping via table7, "File type identification". The extension portion of the filename shall not be used if this descriptor is provided.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

B.3 Stream description

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

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

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

Table 71: Stream description

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

npt_source: A reference to a source of timeline information. This shall be sufficient to derive timeline values, and the timeline rate. It may indicate that no timeline is associated with this stream collection.

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

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

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

audio_stream: True if this stream contains audio.

video_stream: True if this stream contains video.

data_stream: True if this stream contains data.

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

B.4 Trigger signalling

B.4.0 General

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

B.4.1 Trigger object

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

Table 72: Trigger object

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

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

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

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

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

B.4.2 Trigger event

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

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

Table 73: Trigger event

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

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

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

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

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

NOTE 3: A timebase associated with the stream identified by the Trigger object will be used by the terminal to send a trigger to a registered application.

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

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

B.4.2.1 Extrapolation of timebase values

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

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

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

B.4.2.2 Monitoring of trigger events

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

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

B.5 Caching

This clause describes the constraints that a GEM terminal shall implement when caching any content from the object carousel in the memory of the GEM terminal. Caching is optional for the GEM terminal, but if implemented shall conform to the constraints set in this clause.

B.5.1 Determining file version

There is no version number directly related to files (or other BIOP messages), the closest association is the moduleVersion in the DII that references the module that contains the BIOP message. Therefore, to ensure that a file is up to date the GEM terminal must determine that the moduleVersion for the appropriate module is current and reacquire if necessary. The circumstances under which this checking is required are defined by the transparency level as specified in the following clause.

B.5.2 Transparency levels of caching

The definition of transparency levels describes the behaviour that the GEM terminal shall implement when the content in the object carousel is changing. The transparency level determines how certain the GEM terminal is required to be about the validity of the content when returning the content to the application. The object carousel provides a mechanism for determining version changes of the content by monitoring the DII messages.

Validity of content is specified here in terms of the version number of the module that is broadcast in the DII message. The contents of an object as cached in the memory of the GEM terminal are defined to be valid at a certain point in time when the version number of the module in the cache matches the version number of the module as signalled in the DII message describing that module as it was last broadcast.

NOTE: The definition is based on the DII message that was last broadcast and it may be that the GEM terminal was not filtering for this message at that time and did not receive it.

From the GEM terminal point of view, the transparency level indicates the constraints that the terminal needs to implement for monitoring the DII messages.

The broadcaster can indicate the appropriate transparency level that shall be applied for a given piece of content by using a descriptor associated with a module in the DII message. In the absence of this descriptor from a module, the transparent caching is the default level.

B.5.2.1 Transparent caching

The transparent caching is a caching level that ensures that the application can not practically notice a difference in the validity of the returned content between an implementation that caches content and an implementation that does not cache any content. Naturally, an implementation that caches the content will return it to the application faster.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

When returning content from the cache to the application, the GEM terminal shall ensure that the version number of the cached content matches the version number indicated in the current DII message describing that module. Once a DII has been received it can be assumed that it is current at least for 500 ms and after that period until receiving the next instance of the relevant DII. If filtering for that DII has not resumed by the end of this period, the state of that DII is to be considered unknown until it is received again.

Therefore, terminals must not return transparently cached data if it has waited more than half a second between receiving the relevant DII and *starting to filter* for that DII again. If the terminal does not resume filtering within the 500 ms grace period, it must download the relevant DII again when it wishes to use that DII to check cache validity.

The choice of 500 ms is based on the normal timing uncertainty in data delivery through the broadcast chain and is independent of the repetition rate of the DII messages.

B.5.2.1.1 Active caching

There are several ways the GEM terminal can organize its caching strategy. One possible strategy is so-called active caching. This means that the terminal has a dedicated section filter for each DII message it needs to monitor. Keeping that filter continuously filtering for the DII guarantees that the terminal will notice the update of a module as soon as it happens and can thus be aware of the validity of all the content it has cached.

However, in some cases the DII messages might be sent with a very high repetition rate that may cause a high processing load because the terminal needs to do some processing every DII message that it receives. The 500 ms grace period is designed to help this, as it allows the terminal to stop the section filter for 500 ms after receiving the DII message. This lessens the processing burden on the terminal as it only needs to process each DII message twice a second, even if it may be repeated on the transmission much more frequently.

B.5.2.1.2 Passive caching

With active caching, the terminal may need to have a dedicated section filter reserved for each DII message that it needs to monitor. This would effectively limit the amount of content that can be cached, possibly to a very small number. Therefore, the terminal may choose a so-called passive caching strategy. This means that the terminal does not even try to monitor for the DII messages continuously, but each time an application wants to retrieve an object, it at that time retrieves the current DII and checks if the cached content is still valid. Although, this strategy imposes a delay before returning the content to the application, this delay is usually significantly smaller than having to retrieve the content from the broadcast stream.

B.5.2.1.3 DII repetition rate

It should be noted that the description of active and passive caching are only informative here and terminal implementations can use any strategy fulfilling the normative constraints set above. However, broadcasters should set the repetition rate of the DIIs so that a terminal implementing the passive caching strategy will provide the expected benefits of caching over a terminal implementing no caching.

B.5.2.2 Semi-transparent caching

The semi-transparent caching level allows the GEM terminal to cache the data and also return slightly out-dated data to the application. The benefit of this caching level is that it allows terminals to cache larger quantities of content with a reasonable resource usage while allowing the data to be returned usually immediately to the application. The semi-transparent caching level provides less guarantees about validity of the content, but does not cause the delay implied by the passive caching strategy with the transparent caching level.

When returning content from the cache to the application, the terminal shall ensure that the version number of the cached content matches the version number indicated in a valid DII message describing that module. Once a DII has been received it can be assumed to be valid at least for 30 s and after that period until receiving the next instance of the relevant DII. If filtering for that DII has not resumed by the end of this period, the state of that DII is to be considered unknown until it is received again.

Therefore, terminals must not return semi-transparently cached data if it has waited more than 30 seconds between receiving the relevant DII and starting to filter for that DII again. If the terminal does not resume filtering within the 30 s grace period, it must download the relevant DII again when it wishes to use that DII to check cache validity.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

B.5.2.2.1 Implications for the terminal (informative)

Reasons for selecting the 30 s value for the grace period in the semi-transparent caching level are different from the reasons for the 500 ms grace period in the transparent level. The 30 s grace period in this level is intended e.g. to allow terminals to keep typically a valid copy of each DII by retrieving each DII in a round robin fashion using a single section filter. Naturally, whether this goal can be achieved, depends on the repetition rate of the DIIs and the amount of content that is cached. If this is not possible, the terminal might use the passive caching strategy with this transparency level as well. These strategies are only examples and the terminal may implement any strategy as long the normative constraints defined above are fulfilled (this includes implementing no caching as it is optional, as well as treating the semi-transparent level the same as the transparent level).

B.5.2.3 Static caching

When using the static caching transparency level, the GEM terminal shall check the validity of the cached content from the version number in the DII message when it is used for the first time during the lifetime of an application instance. After the first usage time, the GEM terminal does not need to check the validity of the content during the lifetime of that application instance.

B.5.2.3.1 Implications for the broadcaster (informative)

This has the implication, that content with this transparency level is appropriate for very static content that is updated only rarely and where the possible update of the content does not need to be noticed by the application during the lifetime of one application instance.

B.5.2.3.2 Implications for the terminal (informative)

The GEM terminal, however, is allowed to update the contents of the statically cached files if it notices that they have been updated in the carousel as well as use any caching strategy as long as the normative constraint defined above are fulfilled (this includes implementing no caching as it is optional, as well as treating the static level the same as the semi-transparent and/or the transparent level).

B.5.3 Dynamic carousel structure

The Object Carousel may change structure over time, i.e. both files and directories may be added or deleted. Also, modules are not guaranteed to carry the same objects over the lifetime of the carousel. Therefore receivers shall not assume that directory structures are static or that a given path will resolve always to the same object. All cached directory information shall be cached according to the signalled cache priority. This means that before using an object that has been cached, receivers shall validate the path to it.

NOTE: Validating a path does not necessarily mean downloading all elements in the path every time. For example, simply determining that none of the objects on the path have changed since it was last fully traversed is sufficient to confirm that the path itself has not changed.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 347 of 861

Annex C (informative): Bibliography

C.1 Television

UK MHEG Profile, 1.05: "Digital Terrestrial Television MHEG-5 Specification", U.K. DTG.

Porter-Duff, , T. Porter and T. Duff, "Compositing Digital Images", SIGGRAPH 84, 253-259.

E-Book, 1.1, EACEM Technical Report Number TR-030, Baseline Digital Terrestrial TV Receiver Specification.

ETSI ES 202 184 (V1.1.1): "MHEG-5 Broadcast Profile".

ITU-R Recommendation BT.601-5: "Studio Encoding Parameters of Digital Television for standard 4:3 and Wide-Screen 16:9 aspect ratios".

PKCS11, 5, PKCS11 Cryptographic Token Interface Standard.

NOTE: Available at http://www.rsasecurity.com/rsalabs/node.asp?id=2133.

CENELEC EN 50049-1: "Domestic and similar electronic equipment interconnection requirements: Peritelevision connector".

Hunt, R.W.G. (1987): "Measuring Colour (Ellis Horwood Series in Applied Science and Industrial Technology)". Ellis Harwood Limited, Chichester, England.

POSIX, ISBN:1-55937-255-9, IEEE Standard 1003.2-1992: "Standard for Information Technology - Portable Operating System Interfaces (POSIX) - Part 2: Shell and Utilities".

CENELEC EN 50221: "Common interface specification for conditional access and other digital video broadcasting decoder applications".

ETSI TS 101 699 (V1.1.1): "Digital Video Broadcasting (DVB); Extensions to the Common Interface Specification".

R206-001: "Guidelines for Implementation and Use of the Common Interface for DVB Decoder Applications".

ANSI/SCTE 90-1 (2004): "SCTE Applications Platform Part 1 OCAP 1.0 Profile".

IETF RFC 1112 (1989): "Host extensions for IP multicasting".

C.2 Java

Java Media Player guide, 1.03, Nov 6, 1997, Sun Microsystems Java Media Player guide, Java Media Players.

NOTE: Available at http://java.sun.com/products/java-media/jmf/forDevelopers/playerguide/index.html.

Java VM2, ISBN: 0-201-43294-3, The Java Virtual Machine Specification (2nd edition), T. Lindholm and F. Yellin, Addison-Wesley.

Java Class Libraries Vol. 1, ISBN 0-2011002-3, The Java Class Libraries, Second Edition, Volume 1 by Patrick Chan, Rosanna Lee and Douglas Kramer.

Java Class Libraries Vol. 2, ISBN 0-201-31003-1, The Java Class Libraries, Second Edition, Volume 2 by Patrick Chan and Rosanna Lee.

JSR 927, 1.1, Java TV 1.1.

J2SE 5.0, 5, JavaTM 2 Platform Standard Edition 5.0 Development Kit (JDK 5.0).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Java SE, Java SE at a Glance.

NOTE: Available at http://java.sun.com/javase/.

Compilers: Principles, Techniques, and Tools by Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman (Contributor); Addison-Wesley Pub Co, ISBN: 0201100886.

C.3 Internet and HTTP

IETF RFC 2068 (January 1997): "Hypertext Transfer Protocol version 1.1".

IETF RFC 2838 (May 2000): "Uniform Resource Identifiers for Television Broadcasts".

W3C Recommendation 5 May 1999: "Web Content Accessibility Guidelines 1.0".

NOTE: Available at http://www.w3.org/TR/WAI-WEBCONTENT.

W3C Recommendation 26 January 2000: "XHTML™ 1.0 The Extensible HyperText Markup Language".

NOTE: Available at http://www.w3.org/TR/xhtml1.

W3C Recommendation 8 October 2008: "XHTML™ Modularization 1.1.

NOTE: Available at http://www.w3.org/TR/xhtml-modularization.

W3C Recommendation 17 Dec 1996: "Cascading Style Sheets, level 1".

NOTE: Available at http://www.w3.org/TR/REC-CSS1/.

CSS 2 (1998): "Cascading Style Sheets, level 2 CSS2 Specification", Including the errata published in REC-CSS2-19980512.

NOTE: Available at http://www.w3.org/TR/REC-CSS2/.

IETF RFC 2318 (1998): "The text/css Media Type".

W3C Recommendation 29 June 1999: "Associating Style Sheets with XML documents, 1.0".

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

IETF RFC 2109 (1997): "HTTP State Management Mechanism".

IETF RFC 1123 (1989): "Requirements for Internet Hosts Application and Support".

C.4 Other

IETF RFC 1950 (1996): "ZLIB Compressed Data Format Specification version 3.3".

IETF RFC 1951 (1996): "DEFLATE Compressed Data Format Specification version 1.3".

SMIL 1.0.

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

SMIL Boston, "Synchronized Multimedia Integration Language (SMIL)". Boston Specification.

NOTE: Available at http://www.w3.org/TR/smil-boston.

SMIL DOM: "Synchronized Multimedia Integration Language Document". Object Model.

NOTE: Available at http://www.w3.org/TR/smil-boston-dom.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

W3C Recommendation 1 October 1998: "Document Object Model (DOM) Level 1 Specification, V1.0".

NOTE: Available at http://www.w3.org/TR/REC-DOM-Level-1.

W3C Recommendation 13 November 2000: "Document Object Model (DOM) Level 2 Core Specification, V1.0".

NOTE: Available at http://www.w3.org/TR/DOM-Level-2-Core.

W3C Recommendation 13 November 2000: "Document Object Model (DOM) Level 2 Events Specification, V1.0".

NOTE: Available at http://www.w3.org/TR/DOM-Level-2-Events.

W3C Recommendation 13 November 2000: "Document Object Model (DOM) Level 2 Style Specification, V1.0".

NOTE: Available at http://www.w3.org/TR/DOM-Level-2-Style.

W3C Recommendation 13 November 2000: "Document Object Model (DOM) Level 2 Views Specification, V1.0".

NOTE: Available at http://www.w3.org/TR/DOM-Level-2-Views.

W3C Recommendation 16 August 2006: "Namespaces in XML 1.0", World Wide Web Consortium 14-January-1999.

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

W3C Candidate Recommendation 2 August 2000: "Scalable Vector Graphics (SVG) 1.0 Specification".

NOTE: Available at http://www.w3.org/TR/2000/CR-SVG-20000802/.

ETSI TS 102 823: "Digital Video Broadcasting (DVB); Specification for the carriage of synchronized auxiliary data in DVB transport streams".

ETSI TR 101 194 (V1.1.1): "Digital Video Broadcasting (DVB); Guidelines for implementation and usage of the specification of network independent protocols for DVB interactive services".

ITU-T Recommendation X.690: "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)".

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Annex D (normative): Text presentation

D.1 Scope

This clause addresses the following topics:

- How downloaded fonts are associated with applications and accessed by them.
- The DVB-J APIs that are used for presenting text and their behaviour.

Two levels of interface are addressed:

- Simple string rendering as supported by `java.awt.Graphics.drawString`.
- More complex object rendering as supported by DVB Text Layout Manager as described in annex U, "Extended graphics APIs".

Other parts of the present document that are related to this topic are:

- For character sets supported by implementations see annex E, "Character set".
- For the font families, sizes, styles and weights supported by implementations see and the presentation of this to the API clause G.4, "Resident fonts and text rendering".
- For the content formats used to deliver fonts see clause 7.4, "Downloadable fonts".

D.2 Fonts

D.2.1 Embedded fonts

See clause G.4, "Resident fonts and text rendering".

D.2.2 Downloaded fonts

D.2.2.1 Font technology

See clause 7.4, "Downloadable fonts".

D.2.2.2 Font index files

D.2.2.2.1 Format of file

The font index file provides a mapping between a font face name and a file containing the font data. The file syntax is defined by the XML DTD shown in table 74.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 74: Font index file syntax definition

```
<!ELEMENT fontdirectory (font+)>
    <!-- a font definition -->

<!ELEMENT font (name,fontformat,filename,style*,size?)>
    <!-- filename of the font file.
    Because the font directory is per directory, this should
    not contain any directories, but just be a file in that
    directory -->

<!ELEMENT filename (#PCDATA)>
    <!-- font format, e.g. "PFR", "OTF", "TTF" or "TTC" -->

<!ELEMENT fontformat (#PCDATA)>
    <!-- symbolic name of the font -->

<!ELEMENT style (#PCDATA)>
    <!-- font style -->

<!ELEMENT name (#PCDATA)>

<!ELEMENT size EMPTY>
<!ATTLIST size
    min CDATA "0"
    max CDATA "maxint"
>
```

The `PublicLiteral` to be used for specifying this DTD in document type declarations of the XML files is:

```
"-//DVB//DTD Font Directory 1.0//EN"
```

and the URL for the `SystemLiteral` is:

```
"http://www.dvb.org/mhp/dtd/fontdirectory-1-0.dtd"
```

The `Name` used in the document type declaration shall be "fontdirectory".

D.2.2.2.2 Element semantics

font: There shall be one font element per font file included in the font directory.

name: Contains the font face name of the font (e.g. "Helvetica").

fontformat: The file format of the font. For font formats used in the present document, this shall be the following:

- For OpenType fonts with TrueType outlines, this shall be "TTF" or "OTF".
- For OpenType fonts with TrueType collections, this shall be "TTC" or "OTF".
- For PFR fonts, this shall be "PFR".

filename: Relative path to the font file. This is relative to the directory containing the font index file. The separator character for directories is "/". As this is a relative path, it shall not begin with a "/" character.

style: The style elements contain the names of the styles of the font that are contained in this font file. The possible values for GEM are "PLAIN", "BOLD", "ITALIC" and "BOLD_ITALIC". There is one style element included per style contained in the indicated font file, except when all the usable styles of the font are in the same file in which case the style elements can be left out. When different styles of the font are contained in separate files, these are included in the directory as separate font entities with the same name but different style and filename.

size: Indicates the size range for which this font file can be used. The min. attribute contains the minimum size in points (default is "0"). The max attribute contains the maximum size in points or "maxint" if the maximum size is not limited (default is "maxint").

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

If all the usable sizes of the font are generated using the same font file, the size element can be left out. If there are separate files for different sizes, these are included in the directory as separate font entities with the same name and style but different size definition and filename.

D.2.2.2.3 Example

Table 75: Example index file

```
<?xml version="1.0"?>

    <!DOCTYPE fontdirectory PUBLIC "-//DVB//DTD Font Directory 1.0//EN"
    "http://www.dvb.org/mhp/dtd/fontdirectory-1-0.dtd">
    <fontdirectory>
        <font>
            <name>Tiresias</name>
            <fontformat>PFR</fontformat>
            <filename>tiresias.pfr</filename>
            <style>BOLD</style>
        </font>
        <font>
            <name>Broadcaster X Screen Font</name>
            <fontformat>PFR</fontformat>
            <filename>brxsf.pfr</filename>
        </font>
    </fontdirectory>
```

D.2.2.3 Name and location of font index files

D.2.2.3.1 General

The specification of font paths and fonts by an application are private to that application, they are not available to other applications.

D.2.2.3.2 Name of file

The file name shall be:

```
"dvb.fontindex"
```

D.2.2.3.3 Location

The font index file shall be placed in the base directory of the application.

D.2.2.4 Specification of fonts at run time

D.2.2.4.1 DVB-J

The implementation locates the font file using the parameters passed to `org.dvb.ui.FontFactory.createFont()`.

The font file is located by searching the directory for a file where the name element matches the name parameter of a call to `FontFactory.createFont`, a `style` element matches the style parameter and the `size` parameter is within the limitations in the size element.

It is font format specific how the font information for a given name, style and size is encoded in the font file. The name and style need to match the corresponding entry in the font file. The font size needs to be within the range supported by the font file.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

D.3 Text rendering

D.3.1 Low and high level rendering

Two levels of interface are addressed:

D.3.1.1 Low level rendering

Simple string rendering as supported by:

- `java.awt.Graphics.drawString`
- `java.awt.Graphics.drawChars`
- `java.awt.Graphics.drawBytes`

This is referred to as "low level" rendering implying that the application author has significant responsibilities for ensuring that the text is visible. This rendering obeys the normal AWT rules. For example, the author is responsible for placing individual words or lines of text on to a component.

`org.havi.ui.HDefaultTextLayoutManager` shall also be considered as supporting low level rendering except that implementations shall respect the rendering settings on the `HVisible` passed as argument to the render method (such as alignment, font and foreground colour).

D.3.1.2 High level rendering

More complex object rendering as supported by:

- `org.dvb.ui.DVBTextLayoutManager`

This is referred to as "high level" rendering implying that the application author may need less effort to ensure that the text is visible. For example, the author could use the text layout manager to handle flowing paragraphs of text into an `org.havi.ui.HText` object.

D.3.2 Philosophy

This clause describes "logical" rules that ensure text flows predictably on all receivers and defines some rendering requirements to ensure that a minimum acceptable level of text legibility is achieved. The scope of this is much broader for "high level" rendering, where rules address text flow in addition to just string rendering.

No restriction is placed on the rendering technology used in a receiver provided that it achieves the deterministic text flow characteristics and the minimum rendering requirements described in this clause.

D.3.2.1 High level rendering conceptual process

The conceptual rendering process can be described as follows:

a) Based on:

- The size of the object to render into.
- The characteristics of the font (e.g. see clause D.3.3.1, "Font bounds").
- Any automatic or application controlled insets (see clause D.3.5, "Rendering within limits and insets").

Calculate:

- The maximum number of lines of text that may be rendered (see clause D.3.8.1, "Number of lines").
- The width available for rendering on each line (see clause D.3.9.1, "Available width").

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

b) Determine how the text to render flows, effectively defining a series of lines to render using:

- The "logical" rules for calculating the width of rendered text (see clause D.3.6, ""logical" text width rules").
- The available width for rendering (see clause D.3.9.1, "Available width").
- The rules for breaking text (see clause D.3.7, "Line breaking").

c) Determine where each line of text to render is placed vertically within the object (see clauseD.3.8, "Positioning lines of text vertically within an object"). This needs to consider that:

- If the number of lines of text to render (from step b) exceeds the maximum number of lines of text that may be rendered (from step a) "vertical truncation" may be required, i.e. discard some of the lines to render.
- The positioning of lines to render is affected by the vertical justification setting of the object.

d) Determine the placement of individual characters in a line to render (see clause D.3.9, "Rendering lines of text horizontally"). This needs to consider that:

- Even having correctly applied the rules relating to the flow of text, in some extreme circumstances the length of the line of text to render can be wider than the available width for rendering (from step a). To handle this "horizontal truncation" may be required, i.e. discard some of the characters from the line to render.
- The placement of characters in the line to render is affected by the horizontal justification setting of the object.
- The placement of characters in the line to render should ensure sensible and consistent spacing between adjacent characters (see clause D.3.12, "Placing runs of characters and words").

In addition special rules exist for handling tabulation (clause D.3.11, "Tabulation") which need to be taken into account in the implementation of many of these steps.

Behaviour equivalent to the above is required to ensure conformant rendering. The order of these steps is illustrative and may vary between implementations for reasons of convenience or efficiency.

D.3.3 Font Definition

The nomenclature used in this clause is derived from the resident font format(s). The nomenclature and the numerics provided here are directly applicable to downloaded fonts.

D.3.3.1 Font bounds

The font definition induces a set of parameters `xMin`, `xMax`, `yMin` and `yMax` that are properties of the font. These define the maximum extent of the outline representation of characters within the physical font, and as such are defined in terms of outline resolution units (outlineResolution).

(`xMin`, `yMin`) and (`xMax`, `yMax`) are the bottom-left and top-right corners of an imaginary bounding rectangle within which all characters in the font can be completely enclosed.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

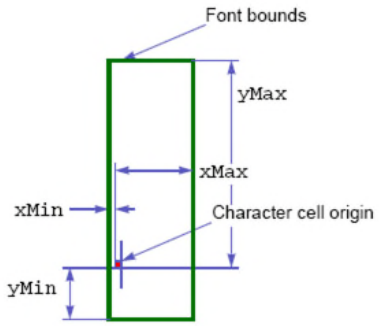


Figure 37: Font bounds

In "Low level rendering" the author is responsible for using knowledge of these values to correctly position text.

In "High level rendering" the text layout manager uses this information when managing text flow to guarantee that the extremities of all characters are completely within the object. See clause D.3.5, "Rendering within limits and insets".

D.3.3.2 "Physical" font data

"Physical" font data such as horizontal escapement and kerning is defined in font metrics resolution units. This is a high-resolution representation, abstracted from any actual rendering system.

In OpenType, the same font resolution units are used for font metric (metric resolution) and outline coordinates (outline resolution), the value is defined by `unitsPerEm` value in the Font Header table. In PFR fonts, the font metrics resolution is defined by the `metricsResolution` field of the Physical Font Record.

NOTE: The `outlineResolution` and `metricsResolution` defined in the PFR are not necessarily the same.

D.3.3.3 Ligatures and Glyph Substitution

When ligatures and glyph substitution are not required by the font, GEM terminals shall not automatically transform letter pairs to ligatures. If ligatures and glyph substitution are supported by the font format, GEM terminals shall automatically select appropriate glyph forms and transform character combinations to ligatures.

GEM applications that utilise PFR fonts and wish to see ligatures rendered should use the appropriate Unicode value (if available) and shall ensure that the glyph associated with this Unicode character is found in the font being used.

D.3.4 Converting font metrics to display pixels

Many of the calculations in this clause are in a high resolution physical coordinate system, either metrics or outline resolutions. These values need to be converted into the pixel resolution of the HGraphicsDevice to allow characters to be rendered.

Values in terms of these high level resolutions can be simply converted to values in terms of points by multiplying by the font size (in points) and dividing by the resolution, i.e. metricsResolution or outlineResolution as appropriate. However, this value in points still needs to be converted into a value in pixels.

Computer display systems typically assume a 72 pixel per inch display. So, as each point is 1/72 inch, the horizontal and vertical size of each pixel is 1 point.

D.3.4.1 Vertical resolution

Each pixel in the graphics device (HGraphicsDevice for DVB-J applications) containing the component is equivalent to a single point.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

D.3.4.2 Horizontal resolution

The horizontal relationship depends on the characteristics of the graphics device (HGraphicsDevice for DVB-J applications). For a square pixel graphics device the 1 pixel = 1 point convention can be preserved.

However, for a graphics device whose pixel aspect ratio is given by `org.havi.ui.HScreenConfiguration.getPixelAspectRatio` the horizontal resolution is the pixel aspect ratio x1 point.

The following table defines the pixel aspect ratios which shall be used when converting typographic pixels for rendering in the graphics system of a GEM terminal for each graphics device aspect ratio.

Table 76: Pixel width for non-square pixel graphics devices

Graphics device resolution	Graphics device aspect ratio	Typographic pixel size width in points
720 x 480	4:3	40/45
	14:9	47/45 (see note)
	16:9	53/45 (see note)
720 x 576	4:3	48/45
	14:9	56/45
	16:9	64/45
NOTE: Approximated value.		

Table 76 is not required for GEM terminal specifications that do not support a graphics device resolution of 720x576, however a functional equivalent shall be specified in GEM terminal specifications.

An emulated graphics could be constructed with 14:9 aspect ratio. This could be used where text is required to be acceptable when viewed on either a 4:3 or 16:9 display. The 14:9 aspect ratio is an artificial construct intended to allow easier authoring but text will always be slightly distorted when displayed. 14:9 should be used when maintaining character aspect ratio is less important to the application than text occupying the same number of pixels regardless of the display size which the GEM terminal is using. A possible example of this is illustrated in figure 38.

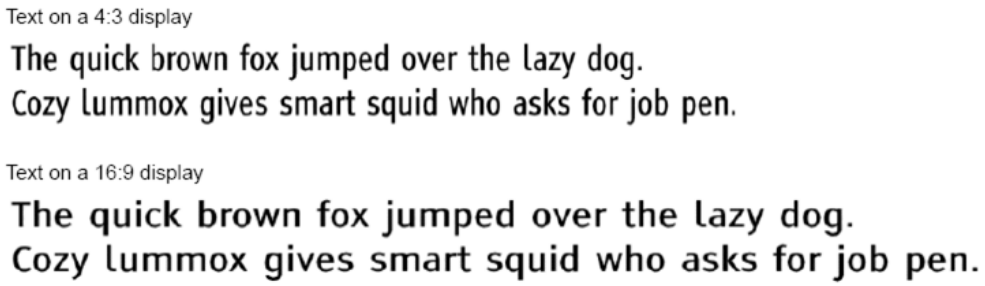


Figure 38: Example of 14:9 text on either 4:3 or 16:9 display

D.3.5 Rendering within limits and insets

When typesetting for print, character extremities may extend beyond the nominal text flow area. However, print has margins so the edge of the text flow is not the technical limit to the area that can be printed.

D.3.5.1 Low level rendering

In "Low level rendering" the author is responsible for placing the text so that it is not clipped.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

D.3.5.2 High level rendering

In "High level rendering" text is automatically rendered with at least a sufficient inset from the object edge (derived from the font properties `xMin`, `yMin`, `xMax` and `yMax`) to ensure that all presented characters are completely rendered within the bounds of the object. Applications can increase or decrease this minimum (font property determined) inset in the following ways:

- The `Insets` parameter on the `org.dvb.ui.DVBTextLayoutManager.render` method.
- The `org.dvb.ui.DVBTextLayoutManager.setInsets` method.

Each of these mechanisms can be used independently. Their effect is cumulative.

For simplicity, these application controlled insets are not described in the specification clauses that follow. In relation to the rendering process their behaviour is equivalent to a reduction in the size of the object within which the rendering processes acts.

D.3.5.3 Conversion of units

As stated previously, these parameters are defined in outline resolution units and so need to be converted to device pixels. Based on the principles described previously (see clause D.3.4, "Converting font metrics to display pixels") this can be achieved by the following:

D.3.5.3.1 yOffsetTop

$$yOffsetTop_{pixels} = \begin{cases} div(yMax_{outlineResolution} \times fontSize, outlineResolution) & yMax > 0 \\ 0 & yMax \leq 0 \end{cases}$$

D.3.5.3.2 yOffsetBottom

$$yOffsetBottom_{pixels} = \begin{cases} div(-yMin_{outlineResolution} \times fontSize, outlineResolution) & yMin < 0 \\ 0 & yMin \geq 0 \end{cases}$$

D.3.5.3.3 xOffsetLeft

D.3.5.3.3.1 With 14:9 graphics

For 720x576 graphics device resolution:

$$xOffsetLeft_{pixels} = \begin{cases} div(-xMin_{outlineResolution} \times fontSize \times 45, outlineResolution \times 56) & xMin < 0 \\ 0 & xMin \geq 0 \end{cases}$$

For 720x480 graphics device resolution:

$$xOffsetLeft_{pixels} = \begin{cases} div(-xMin_{outlineResolution} \times fontSize \times 45, outlineResolution \times 47(56)) & xMin < 0 \\ 0 & xMin \geq 0 \end{cases}$$

D.3.5.3.3.2 With 4:3 graphics

For 720x576 graphics device resolution:

$$xOffsetLeft_{pixels} = \begin{cases} div(-xMin_{outlineResolution} \times fontSize \times 45, outlineResolution \times 48) & xMin < 0 \\ 0 & xMin \geq 0 \end{cases}$$

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

For 720x480 graphics device resolution:

$$xOffsetLeft_{pixels} = \begin{cases} div(-xMin_{outlineResolution} \times fontSize \times 45, outlineResolution \times 40) & xMin < 0 \\ 0 & xMin \geq 0 \end{cases}$$

D.3.5.3.3.3 With 16:9 graphics

For 720x576 graphics device resolution

$$xOffsetLeft_{pixels} = \begin{cases} div(-xMin_{outlineResolution} \times fontSize \times 45, outlineResolution \times 64) & xMin < 0 \\ 0 & xMin \geq 0 \end{cases}$$

For 720x480 graphics device resolution

$$xOffsetLeft_{pixels} = \begin{cases} div(-xMin_{outlineResolution} \times fontSize \times 45, outlineResolution \times 53) & xMin < 0 \\ 0 & xMin \geq 0 \end{cases}$$

D.3.5.3.4 outlineResolution

outlineResolution is extracted from the font data (see clauses 11.4.1.5, "Font bindings" and D.3.3.2, ""Physical" font data"), div (A, B) = ceil (A / B), "/" is a rational divide and ceil(A) is the first integral number greater than or equal to A.

D.3.5.3.5 Font bounds

yOffsetTop, *yOffsetBottom* and *xOffsetLeft* are all greater than or equal to zero and represent the number of available pixels required above, below and to the left of the character origin to prevent clipping of any character in the font. A value *xOffsetRight* could be defined along similar lines but is not relevant to GEM.

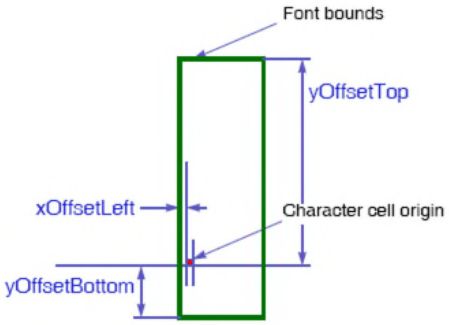


Figure 39: Font bounds

D.3.6 "logical" text width rules

This clause applies to both "Low level rendering" and "High level rendering". Its purpose is to ensure that text will flow predictably on different receivers and authoring stations, regardless of the quality of the character rendering, a set of "logical" text width rules are defined here.

NOTE: I.e. lines and words will break at the same character position.

These rules are a simplification of the rules that might be applied in a typographic rendering system. The objective of these simplifications is to reduce the receiver complexity required to ensure exact correlation of text flow behaviour.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

The calculation of "logical" text width is based on "physical" font data. This data provides a description of the font at a very high resolution, abstracted from any actual rendering system. Consequently, the calculation of the "logical" width of a string of characters involves, computing their width at this high resolution and then converting to units appropriate to the rendering system, e.g. device pixels, before making decisions about text flow (see clause D.3.4, "Converting font metrics to display pixels").

In the case of "Low level rendering" it defines the internal computation performed by the AWT routines that measure the width of text:

- `java.awt.FontMetrics.charWidth`.
- `java.awt.FontMetrics.stringWidth`.
- `java.awt.FontMetrics.bytesWidth`.

Due to the rounding processes within the calculations invoking these methods on subsets of a string may not yield the same total result as invoking the methods on the complete string. In particular the total of the values returned by `java.awt.FontMetrics.getWidths` may be different from the value returned by `java.awt.FontMetrics.stringWidth` for the same string.

In the case of "High level rendering" it defines the computations that the layout manager uses in the following cases:

- To determine when to wrap lines of text within an object.
- To determine which tab stops text has passed when implementing tab characters.

D.3.6.1 Computing "logical" text width

The key parameters when calculating the width of a string of N characters are:

- Text font size.
- charSetWidth.
- The metricsResolution.
- Any kerning adjustment.

D.3.6.1.1 Font sizes

Font sizes are expressed as the size of an "Em" in units of "points".

NOTE 1: Broadly speaking an Em is the minimum distance between the baselines of consecutive lines of text in the given font. If text is 48 point then the Em at that size is 48 points.

NOTE 2: The point is an archaic typographical unit. Traditionally there were 72,27 points to an inch. Computerized systems now use 72 points per inch for simplicity.

D.3.6.1.2 Character widths

The font definition gives the width of each character relative to the size of an Em in metricsResolution units.

NOTE: If metrics are specified in 1/1 000 ths of an Em a character with a width of 0,6 Em will have a set width of 600.

D.3.6.1.3 Kerning

For certain character combinations (a "kerning pair") a kerning adjustment may also be provided. Typically kerning reduces character spacing for pairs such as AV instead of A V, these provide a signed adjustment to the nominal charSetWidth of the first character.

Like charSetWidth kerning adjustments are in terms of metricsResolution units.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Kerning adjustments only apply between non whitespace characters, not between the start of a line of text and the edge of the object. If justification is being used, only whitespace between words may be adjusted.

D.3.6.1.4 Letter spacing

Letter spacing allows for expansion/condensation of the character spacing for all of the characters in a object including whitespace.

NOTE: In printing, this is sometimes referred to as tracking.

D.3.6.2 Logical text width

The equation below shows how the width of a string of N characters is computed.

$$\text{logical width of N characters}_{\text{points}} =$$

$$div((N-1)\times \text{letterspace}, 256) \div div\left(\text{fontsize}\times\left(\sum_{1}^{N}\text{charSetWidth}\left[\right]+\sum_{1}^{N-1}\text{kern}\left[i+1\right]\right), \text{metricsResolution}\right)$$

$$\text{logical width of N characters}_{\text{pixels}} = div\left(\text{logical width of N characters}_{\text{points}}\times H, W\right)$$

Where in div(A, B):

- B is unsigned and A is signed; and
- div(A, B) = ceil(A / B).

Where '/' is a rational divide and ceil(A) is the first integral number greater than or equal to A. So, the calculations round up when reducing precision and tend to over estimate the width of text.

Where H and W are respectively the height and width returned by `org.havi.ui.HScreenConfiguration.getPixelAspectRatio()`.

D.3.7 Line breaking

D.3.7.1 Text wrapping setting is false

When the text wrapping setting is false, text shall break onto a new line only where a Carriage Return character is present in the text. The number of lines to render shall be equal to the number of Carriage Returns present, plus one.

NOTE: The "plus one" ensures that the last line of any text can be presented even if it has not been terminated with a Carriage Return.

D.3.7.2 Text wrapping setting is true

When the text wrapping setting is true, the text flow may break on to new lines at positions in addition to those caused by Carriage Return characters. These additional break positions are defined below. This behaviour is independent of the object's horizontal alignment settings and is considered to take place before any truncation (see clause D.3.8.2, "Truncation").

NOTE 1: This clause may not apply for non-western languages. GEM terminal specifications that support languages for which this clause does not apply, provide a functional equivalent.

The effect of text wrapping on the rendering process is equivalent to substituting "breaking characters" (defined in table 77 "Special characters") with Carriage Return characters at positions determined by the wrapping rules.

The behaviour of the wrapping process is described below:

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 361 of 861

a) Based on the "logical" width of the text, receivers shall determine for each line, the first contiguous sequence of non-breaking characters that meets both of the following requirements:

- Would not completely fit within the available width (see clause D.3.9.1, "Available width").
- That follows one or more breaking characters.

If such a sequence exists, the breaking character preceding it shall be replaced with a Carriage Return character.

b) All trailing breaking characters shall be discarded from all lines of text. (Preceding breaking characters are not affected).

NOTE 2: In general use this identifies the word in a body of text that if rendered would cause the current line to spill outside of the text object. This word becomes the first word in the next line. However, if a line starts with a single word that is longer than the width of the object then the line is broken just before a following word.

EXAMPLE 1: This illustrates the most common case where the word "won't" does not fit and so is broken to the next line.

EXAMPLE 2: This illustrates the particular case of a long first word. In this case "is" is the first word that does not fit AND is preceded by a breaking character.

EXAMPLE 3: This illustrates the consumption of trailing spaces. The line breaks just before the second "spaces" (so the next line starts with this word) and all trailing spaces (e.g. after the first "spaces,") are discarded).

NOTE 3: The importance of the removal of trailing breaking characters is particularly visible if the text is centred or right justified. For example, the text to be centred becomes "spaces" rather than " spaces".

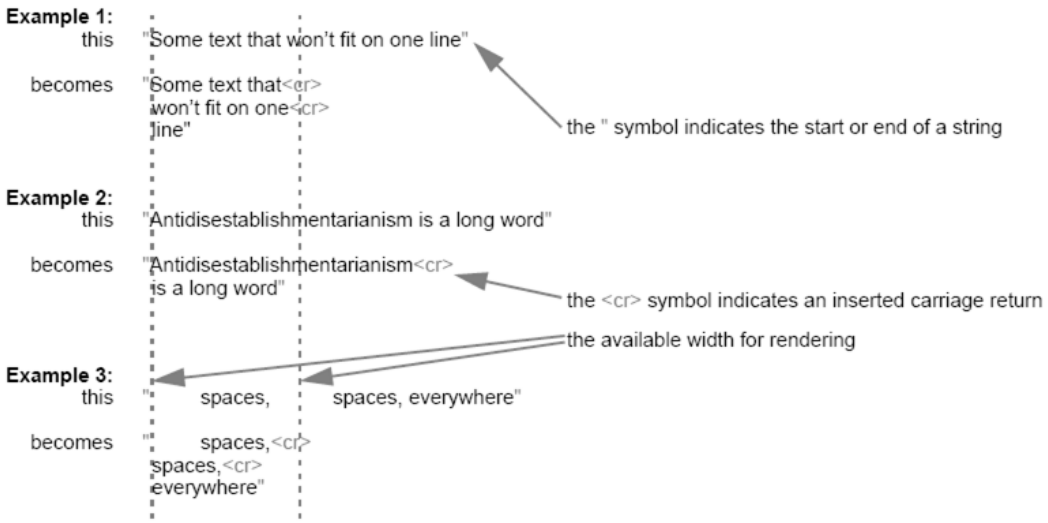


Figure 40: Text wrapping examples

After text wrapping the text truncation rules have to be considered. For example, in example 2 in figure 40 the word "Antidisestablishmentarianism" will be truncated (see clause D.3.9.2, "Truncation").

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

D.3.8 Positioning lines of text vertically within an object

D.3.8.1 Number of lines

Assuming that all characters in a line of text share a common baseline then, regardless of the vertical alignment setting the number of lines of text that can be presented within an object is:

```
num_lines = floor((object_height - (yOffsetBottom + yOffsetTop)) / linespace) + 1
```

All values are in pixels.

object_height: is the height of the component less any margin.

linespace: is an attribute of the object that defines the space between the baselines of consecutive lines of text. This is defined in units of points but is converted to pixels (see clause D.3.4, "Converting font metrics to display pixels").

The function floor(A) rounds A to the first integral number less than or equal to A.

D.3.8.2 Truncation

When the number of lines of text to be rendered exceeds the available height within the object then lines of text shall be discarded as follows:

- If the vertical alignment setting is VERTICAL_CENTER or VERTICAL_START_ALIGN then lines are discarded from the end of the text.
- If the vertical alignment setting is VERTICAL_END_ALIGN then lines are discarded from the beginning of the text.

This truncation will result in a number of lines that can be completely displayed within the object. These shall be presented according to clause D.3.8.3, "Positioning".

See also clause D.3.10, "Text overflow".

D.3.8.3 Positioning

The origin of any character shall be at least yOffsetTop inside the top edge of the object and at least yOffsetBottom inside its bottom edge. The spacing between the baselines of text shall be `linespace`.

D.3.8.3.1 Vertical alignment setting is VERTICAL_START_ALIGN

When the text is top aligned the baseline of the top most line shall be yOffsetTop below the top of the object.

D.3.8.3.2 Vertical alignment setting is VERTICAL_END_ALIGN

When the text is bottom aligned the baseline of the bottom most line shall be yOffsetBottom above the bottom of the object.

D.3.8.3.3 Vertical alignment setting is VERTICAL_CENTER

When the text is vertically centred the positioning of the lines shall be such that the space above the first line of text equals the space below the last line of text (to allow for rounding the lower gap may be up to one pixel greater than the upper gap).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

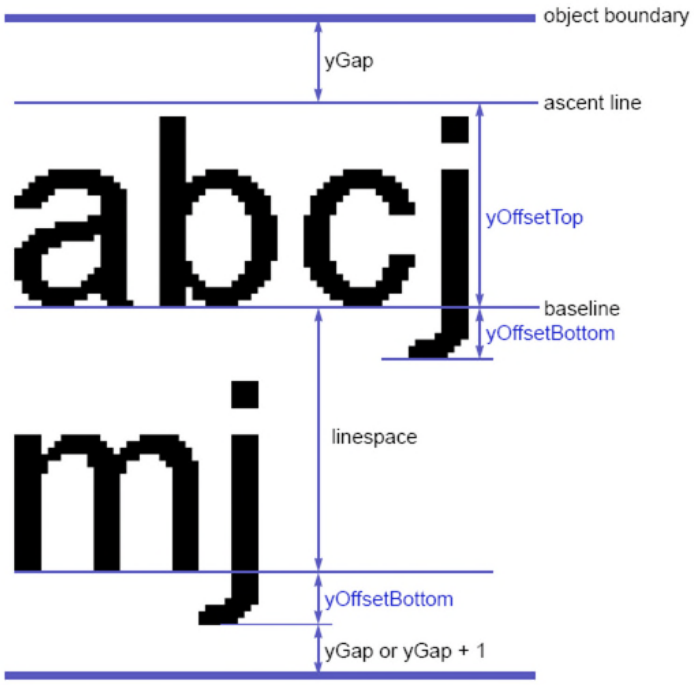


Figure 41: Vertical measures

D.3.8.3.4 Examples

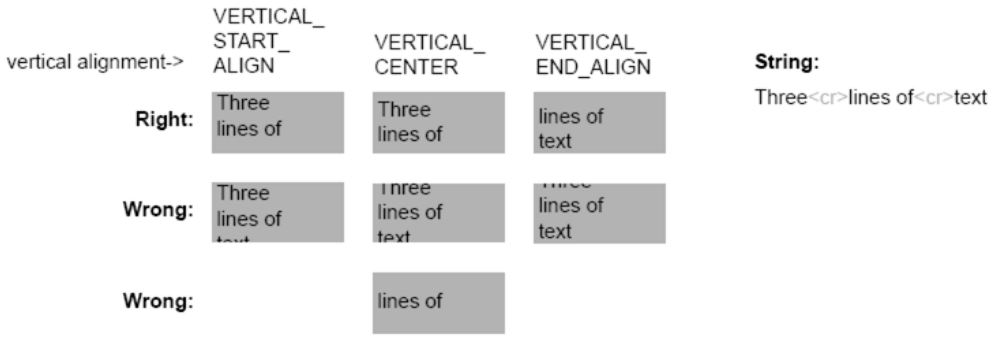


Figure 42: Vertical positioning example 1

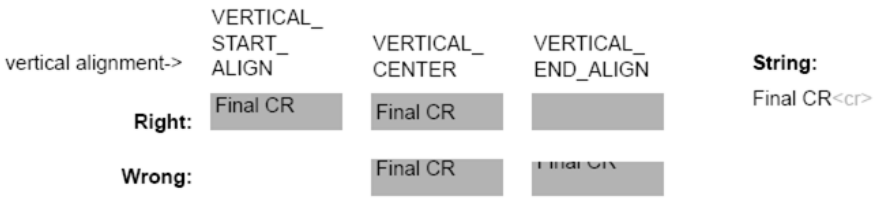


Figure 43: Vertical positioning example 2

D.3.9 Rendering lines of text horizontally

D.3.9.1 Available width

The number of characters that may be rendered on a line is not simply dependent upon the width of the object and the horizontal escapement for each character, but also needs to consider that the rendering of the first character in a line may extend to the left of its origin. Thus, regardless of the horizontal alignment setting the space available for rendering a line of text within an object is:

```
available_width = object_width - xOffsetLeft
```

All values are in pixels. This derivation of `available_width` shall be used with the "logical" text width rules to determine text flow.

D.3.9.2 Truncation

Where a line of text is too long to fit within the `available_width` of the object, the line shall be truncated.

NOTE: This addresses the case where text wrapping has not made the text fit. This can happen either because the text wrapping setting is false, or because a single word is too long to fit.

The result shall be as if the complete line were aligned appropriately on the object (taking into account the horizontal alignment setting) with only those characters which can be completely presented being rendered. That is those characters:

- Whose origin is more than xOffsetLeft right of the left hand edge of the object.
- Whose right hand edge falls inside the right hand edge of the object.

For example:

- If the horizontal alignment setting is `HORIZONTAL_END_ALIGN` then a portion of text from the right end of the line will be rendered with its edge aligned to the right edge of the object.
- If the horizontal alignment setting is `HORIZONTAL_CENTER` the centre of the string will appear at the centre of the object with excess characters being truncated from each end.

See also clause D.3.10, "Text overflow".

D.3.9.3 Placement

- When the horizontal alignment setting is `HORIZONTAL_START_ALIGN` the origin of the left most character shall be xOffsetLeft inside the left edge of the object.
- When the horizontal alignment setting is `HORIZONTAL_CENTER` the positioning of the characters shall be such that the gap to the left of the text equals the gap to the right (to within one pixel).
- When the horizontal alignment setting is `HORIZONTAL_END_ALIGN` the origin of the right most character shall be as necessary to ensure that it is completely visible when rendered.

NOTE 1: xOffsetLeft allows for characters (such as capital "J") that extend to the left of their origin. xOffsetLeft is a property of the font and so represents a consistent indent, i.e. it is independent of the first character.

NOTE 2: There are characters (e.g. fractional divisor) that can extend beyond the escapement indicated by their charSetWidth value. These can theoretically extend beyond the right hand edge of the object if the character is at the end of a word and that word is at the end of a line and the "logical width" of the line just fits the object. With the "fractional divisor" character this is a very unlikely combination of circumstances as this character is normally embedded within a "word". The handling of cases such as this by the GEM terminal is implementation dependent.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

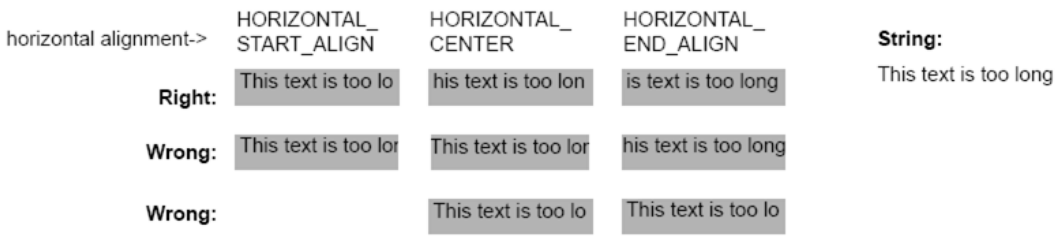


Figure 44: Horizontal positioning examples

D.3.10 Text overflow

The `org.dvb.ui.TextOverflowListener.notifyTextOverflow` method is called if truncation occurs.

NOTE: This can be used by an application to alert the viewer or take other remedial action.

D.3.11 Tabulation

In left aligned text (the horizontal alignment setting is `HORIZONTAL_START_ALIGN`) tab stops are defined at intervals of `horizontalTabSpace` from the left edge of the object. By default the value of `horizontalTabSpace` is 56 points. "Horizontal Tabulation" advances the origin of the next character to be rendered to the next tab stop in the direction that the text is currently flowing (the character repertoire in Table 81 only requires left to right text).

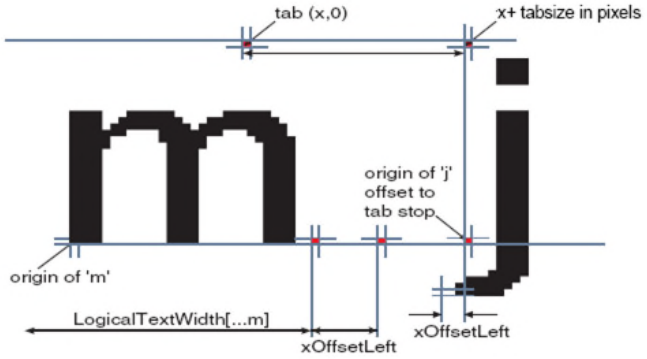


Figure 45: Effect of horizontal tabulation

- Tab characters only have meaning in left aligned text (the horizontal alignment setting is `HORIZONTAL_START_ALIGN`). If the text is right aligned, centred or justified then tab character shall be treated as a space character.
- A tab logically advances the rendering of the text by at least the width xOffsetLeft. If the normal origin of the next character to be rendered after the tab character is after a tab stop, a tab character will advance the rendering to the subsequent tab stop.
- The tab stops are at regular intervals from the left edge of the object and are not affected by the xOffsetLeft offset to the origin of the first character.
- The tab size of 56 points shall be converted to pixels using table 76.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

D.3.12 Placing runs of characters and words

A run of characters starts from a well defined point:

- The start edge of the object.
- A tab stop.

After this origin the fine positioning of character cells and the gaps between words is not fully specified.

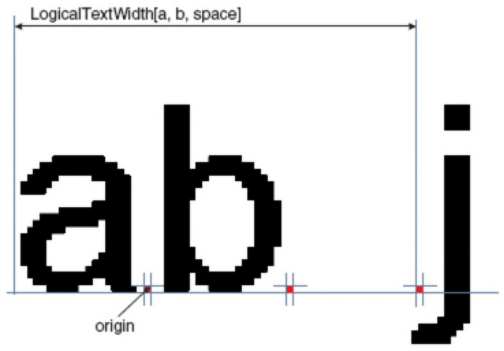


Figure 46: Calculation of character placement

However, the following rendering requirements shall be observed to ensure that a minimum acceptable level of text legibility is achieved:

- The spacing between any pair of non whitespace characters should be consistent wherever that pair of characters is displayed.
- At the default character spacing no two non-whitespace characters should "appear" to touch, except for special cases such as where ligatures dipthongs etc. are being used. The definition of the underlying font may make this requirement impossible.
- The physical rendering of a run of text as determined by the "logical" rules shall be achieved completely within the space used for the "logical" calculation.
- No partially rendered characters shall be presented.

NOTE: The "logical" position of each character as determined by the "logical" rules is likely to be a non-integer. For renderers unable to support suitable sub-pixel positioning, e.g. a 1-bit/pixel bitmap renderer, this is impossible to implement. Thus, whilst the "logical" rules are used to determine the flow of text, they are not required to be observed for individual character placement within this flow and other strategies may be employed as long as the requirements above are satisfied.

D.3.13 Control of text flow

See the definition of the DVB text layout manager.

D.4 Text mark-up

This clause on text mark-up applies to "High level rendering".

D.4.1 White Space Characters

Certain non-printing characters have special meaning. These are identified in table 77.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 77: Special characters

UTF8 Value(s)	Character	Name	Breaking/ Non-breaking	Meaning
0x09	0x0009	Tab	Breaking	See clause D.3.11, "Tabulation".
0x0A	0x000A	Carriage Return	N/A	Causes the text flow to break. The origin for the next character to be rendered moves to a new baseline "linespace" below that just rendered. The horizontal position of the next line will depend on the horizontal alignment setting.
0x0D	0x000D			
0x0D0A (see note)	0x000D 0x000A			
0x20	0x0020	Space	Breaking	Spaces text by the width defined for the space character. When an object has its text wrapping setting set to "true", lines may be broken at a space. See clause D.3.7, "Line breaking".
0xC2A0	0x00A0	Non-breaking Space	Non-breaking	Identical spacing characteristics to 0x20 but is not seen as word boundary for deciding a position to break a line of text. (0xC2A0 is the UTF-8 representation of 0x00A0)
0xE28087	0x2007	Figure Space	Non-breaking	Can be used in a string of numerals as an alternative to using comma to denote "thousands". This character is not treated as a word boundary when deciding a position to break a line of text.
NOTE: The character sequence 0x0D0A shall be rendered identically to a single 0x0D.				

D.4.2 Marker characters

The codes 0x1C to 0x1F are zero width, non-spacing, non-printing characters available for use by authors as markers in objects, i.e. when using string operations.

D.4.3 Non-printing characters

Certain characters (or character sequences) have no immediate visual representation.

These include:

- 0x1C to 0x1F marker characters (see clause D.4.2).
- Format control mark-up (see clause D.4.4).
- Other characters not recognized by the receiver.

When presenting text that includes these characters the character placement shall be as if the non-printing characters were eliminated from the text before rendering. In particular, the character spacing and inter character kerning shall be computed as if the non-printing characters were not present.

D.4.4 Format Control Mark-up

Within objects mark-up codes can be used to control the presentation of text. The sequence in table 78 marks the start of some marked-up text. For each "start of mark-up" a corresponding "end of mark-up" is defined. The byte sequence for the "end of mark-up" is illustrated in Table 79. The minimum number of supported mark-up instances, where each instance is a start and end mark-up pair, is 256.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 78: General format for start of text mark-up

	Bits	Value	Note
start_of_markup	8	0x1B	Escape
markup_start_identifier	8	0x40 to 0x5E	"@" to "^"
parameters_length	8	N	
for(i=0; i<N; i++) {			
parameter_byte	8	0x00 to 0xFF	
}			

Table 79: General format for end of text mark-up

	Bits	Value	Note
end_of_markup	8	0x1B	Escape
markup_end_identifier	8	0x60 to 0x7E	"`" to "~"

Table 80: Codes defined for use in marked-up text files

Min. Nesting	Start mark-up	End mark-up	Description
	0x1B 0x42 0x00	0x1B 0x62	Applies "bold" style to the text enclosed (see note 1)
16	0x1B 0x43 0x04 0xrr 0xgg 0xbb 0xtt	0x1B 0x63	Applies colour to the text enclosed. 0xrr specifies the red intensity, 0xgg the green, 0xbb the blue and 0xtt the transparency (where 0x00 is fully opaque and 0xff fully transparent).
NOTE 1: Not supported in this version of GEM. NOTE 2: The required text encoding for this mark-up format is described in clause 11.2.11, "Text Encodings".			

D.4.5 Future compatibility

Compatible extensions to the set of mark-up codes may be defined in future versions of GEM. For each the `markup_end_identifier` will be 32 (0x20) greater than the `markup_start_identifier`. GEM terminals shall ignore unrecognized mark-up and shall display any text enclosed within an unrecognized mark-up.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Annex E (normative): Character set

A GEM implementation shall be able to *display* and *accept as input* at least the set of characters shown in table 81.

The range of characters accepted as input may be limited by the capabilities of the available input devices. This clause does not imply that keyboards (real and virtual) for GEM terminals are required to support input of this full character set.

Table 81: Extended Latin set

Code	ISO/IEC 10646-1 [15] Character name
0x0020	SPACE
0x0021	EXCLAMATION MARK
0x0022	QUOTATION MARK
0x0023	NUMBER SIGN
0x0024	DOLLAR SIGN
0x0025	PERCENT SIGN
0x0026	AMPERSAND
0x0027	APOSTROPHE
0x0028	LEFT PARENTHESIS
0x0029	RIGHT PARENTHESIS
0x002A	ASTERISK
0x002B	PLUS SIGN
0x002C	COMMA
0x002D	HYPHEN-MINUS
0x002E	FULL STOP
0x002F	SOLIDUS
0x0030	DIGIT ZERO
0x0031	DIGIT ONE
0x0032	DIGIT TWO
0x0033	DIGIT THREE
0x0034	DIGIT FOUR
0x0035	DIGIT FIVE
0x0036	DIGIT SIX
0x0037	DIGIT SEVEN
0x0038	DIGIT EIGHT
0x0039	DIGIT NINE
0x003A	COLON
0x003B	SEMICOLON
0x003C	LESS-THAN SIGN
0x003D	EQUALS SIGN
0x003E	GREATER-THAN SIGN
0x003F	QUESTION MARK
0x0040	COMMERCIAL AT
0x0041	LATIN CAPITAL LETTER A
0x0042	LATIN CAPITAL LETTER B
0x0043	LATIN CAPITAL LETTER C
0x0044	LATIN CAPITAL LETTER D
0x0045	LATIN CAPITAL LETTER E
0x0046	LATIN CAPITAL LETTER F
0x0047	LATIN CAPITAL LETTER G
0x0048	LATIN CAPITAL LETTER H
0x0049	LATIN CAPITAL LETTER I
0x004A	LATIN CAPITAL LETTER J
0x004B	LATIN CAPITAL LETTER K
0x004C	LATIN CAPITAL LETTER L
0x004D	LATIN CAPITAL LETTER M
0x004E	LATIN CAPITAL LETTER N
0x004F	LATIN CAPITAL LETTER O
0x0050	LATIN CAPITAL LETTER P
0x0051	LATIN CAPITAL LETTER Q

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Code	ISO/IEC 10646-1 [15] Character name
0x0052	LATIN CAPITAL LETTER R
0x0053	LATIN CAPITAL LETTER S
0x0054	LATIN CAPITAL LETTER T
0x0055	LATIN CAPITAL LETTER U
0x0056	LATIN CAPITAL LETTER V
0x0057	LATIN CAPITAL LETTER W
0x0058	LATIN CAPITAL LETTER X
0x0059	LATIN CAPITAL LETTER Y
0x005A	LATIN CAPITAL LETTER Z
0x005B	LEFT SQUARE BRACKET
0x005C	REVERSE SOLIDUS
0x005D	RIGHT SQUARE BRACKET
0x005E	CIRCUMFLEX ACCENT
0x005F	LOW LINE
0x0060	GRAVE ACCENT
0x0061	LATIN SMALL LETTER A
0x0062	LATIN SMALL LETTER B
0x0063	LATIN SMALL LETTER C
0x0064	LATIN SMALL LETTER D
0x0065	LATIN SMALL LETTER E
0x0066	LATIN SMALL LETTER F
0x0067	LATIN SMALL LETTER G
0x0068	LATIN SMALL LETTER H
0x0069	LATIN SMALL LETTER I
0x006A	LATIN SMALL LETTER J
0x006B	LATIN SMALL LETTER K
0x006C	LATIN SMALL LETTER L
0x006D	LATIN SMALL LETTER M
0x006E	LATIN SMALL LETTER N
0x006F	LATIN SMALL LETTER O
0x0070	LATIN SMALL LETTER P
0x0071	LATIN SMALL LETTER Q
0x0072	LATIN SMALL LETTER R
0x0073	LATIN SMALL LETTER S
0x0074	LATIN SMALL LETTER T
0x0075	LATIN SMALL LETTER U
0x0076	LATIN SMALL LETTER V
0x0077	LATIN SMALL LETTER W
0x0078	LATIN SMALL LETTER X
0x0079	LATIN SMALL LETTER Y
0x007A	LATIN SMALL LETTER Z
0x007B	LEFT CURLY BRACKET
0x007C	VERTICAL LINE
0x007D	RIGHT CURLY BRACKET
0x007E	TILDE
0x00A0	NO-BREAK SPACE
0x00A1	INVERTED EXCLAMATION MARK
0x00A2	CENT SIGN
0x00A3	POUND SIGN
0x00A4	CURRENCY SIGN
0x00A5	YEN SIGN
0x00A8	DIAERESIS
0x00A9	COPYRIGHT SIGN
0x00AA	FEMININE ORDINAL INDICATOR
0x00AB	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
0x00AC	NOT SIGN
0x00AD	SOFT HYPHEN
0x00AE	REGISTERED SIGN
0x00B0	DEGREE SIGN
0x00B4	ACUTE ACCENT
0x00B6	PILCROW SIGN
0x00B7	MIDDLE DOT
0x00B8	CEDILLA
0x00BA	MASCULINE ORDINAL INDICATOR

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 371 of 861

Code	ISO/IEC 10646-1 [15] Character name
0x00BB	RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
0x00BC	VULGAR FRACTION ONE QUARTER
0x00BD	VULGAR FRACTION ONE HALF
0x00BE	VULGAR FRACTION THREE QUARTERS
0x00BF	INVERTED QUESTION MARK
0x00C0	LATIN CAPITAL LETTER A WITH GRAVE
0x00C1	LATIN CAPITAL LETTER A WITH ACUTE
0x00C2	LATIN CAPITAL LETTER A WITH CIRCUMFLEX
0x00C3	LATIN CAPITAL LETTER A WITH TILDE
0x00C4	LATIN CAPITAL LETTER A WITH DIAERESIS
0x00C5	LATIN CAPITAL LETTER A WITH RING ABOVE
0x00C6	LATIN CAPITAL LETTER AE
0x00C7	LATIN CAPITAL LETTER C WITH CEDILLA
0x00C8	LATIN CAPITAL LETTER E WITH GRAVE
0x00C9	LATIN CAPITAL LETTER E WITH ACUTE
0x00CA	LATIN CAPITAL LETTER E WITH CIRCUMFLEX
0x00CB	LATIN CAPITAL LETTER E WITH DIAERESIS
0x00CC	LATIN CAPITAL LETTER I WITH GRAVE
0x00CD	LATIN CAPITAL LETTER I WITH ACUTE
0x00CE	LATIN CAPITAL LETTER I WITH CIRCUMFLEX
0x00CF	LATIN CAPITAL LETTER I WITH DIAERESIS
0x00D0	LATIN CAPITAL LETTER ETH
0x00D1	LATIN CAPITAL LETTER N WITH TILDE
0x00D2	LATIN CAPITAL LETTER O WITH GRAVE
0x00D3	LATIN CAPITAL LETTER O WITH ACUTE
0x00D4	LATIN CAPITAL LETTER O WITH CIRCUMFLEX
0x00D5	LATIN CAPITAL LETTER O WITH TILDE
0x00D6	LATIN CAPITAL LETTER O WITH DIAERESIS
0x00D7	MULTIPLICATION SIGN
0x00D8	LATIN CAPITAL LETTER O WITH STROKE
0x00D9	LATIN CAPITAL LETTER U WITH GRAVE
0x00DA	LATIN CAPITAL LETTER U WITH ACUTE
0x00DB	LATIN CAPITAL LETTER U WITH CIRCUMFLEX
0x00DC	LATIN CAPITAL LETTER U WITH DIAERESIS
0x00DD	LATIN CAPITAL LETTER Y WITH ACUTE
0x00DE	LATIN CAPITAL LETTER THORN
0x00DF	LATIN SMALL LETTER SHARP S
0x00E0	LATIN SMALL LETTER A WITH GRAVE
0x00E1	LATIN SMALL LETTER A WITH ACUTE
0x00E2	LATIN SMALL LETTER A WITH CIRCUMFLEX
0x00E3	LATIN SMALL LETTER A WITH TILDE
0x00E4	LATIN SMALL LETTER A WITH DIAERESIS
0x00E5	LATIN SMALL LETTER A WITH RING ABOVE
0x00E6	LATIN SMALL LETTER AE
0x00E7	LATIN SMALL LETTER C WITH CEDILLA
0x00E8	LATIN SMALL LETTER E WITH GRAVE
0x00E9	LATIN SMALL LETTER E WITH ACUTE
0x00EA	LATIN SMALL LETTER E WITH CIRCUMFLEX
0x00EB	LATIN SMALL LETTER E WITH DIAERESIS
0x00EC	LATIN SMALL LETTER I WITH GRAVE
0x00ED	LATIN SMALL LETTER I WITH ACUTE
0x00EE	LATIN SMALL LETTER I WITH CIRCUMFLEX
0x00EF	LATIN SMALL LETTER I WITH DIAERESIS
0x00F0	LATIN SMALL LETTER ETH
0x00F1	LATIN SMALL LETTER N WITH TILDE
0x00F2	LATIN SMALL LETTER O WITH GRAVE
0x00F3	LATIN SMALL LETTER O WITH ACUTE
0x00F4	LATIN SMALL LETTER O WITH CIRCUMFLEX
0x00F5	LATIN SMALL LETTER O WITH TILDE
0x00F6	LATIN SMALL LETTER O WITH DIAERESIS
0x00F7	DIVISION SIGN
0x00F8	LATIN SMALL LETTER O WITH STROKE
0x00F9	LATIN SMALL LETTER U WITH GRAVE
0x00FA	LATIN SMALL LETTER U WITH ACUTE

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 372 of 861

Code	ISO/IEC 10646-1 [15] Character name
0x00FB	LATIN SMALL LETTER U WITH CIRCUMFLEX
0x00FC	LATIN SMALL LETTER U WITH DIAERESIS
0x00FD	LATIN SMALL LETTER Y WITH ACUTE
0x00FE	LATIN SMALL LETTER THORN
0x00FF	LATIN SMALL LETTER Y WITH DIAERESIS
0x0100	LATIN CAPITAL LETTER A WITH MACRON
0x0101	LATIN SMALL LETTER A WITH MACRON
0x0102	LATIN CAPITAL LETTER A WITH BREVE
0x0103	LATIN SMALL LETTER A WITH BREVE
0x0104	LATIN CAPITAL LETTER A WITH OGONEK
0x0105	LATIN SMALL LETTER A WITH OGONEK
0x0106	LATIN CAPITAL LETTER C WITH ACUTE
0x0107	LATIN SMALL LETTER C WITH ACUTE
0x0108	LATIN CAPITAL LETTER C WITH CIRCUMFLEX
0x0109	LATIN SMALL LETTER C WITH CIRCUMFLEX
0x010A	LATIN CAPITAL LETTER C WITH DOT ABOVE
0x010B	LATIN SMALL LETTER C WITH DOT ABOVE
0x010C	LATIN CAPITAL LETTER C WITH CARON
0x010D	LATIN SMALL LETTER C WITH CARON
0x010E	LATIN CAPITAL LETTER D WITH CARON
0x010F	LATIN SMALL LETTER D WITH CARON
0x0110	LATIN CAPITAL LETTER D WITH STROKE
0x0111	LATIN SMALL LETTER D WITH STROKE
0x0112	LATIN CAPITAL LETTER E WITH MACRON
0x0113	LATIN SMALL LETTER E WITH MACRON
0x0116	LATIN CAPITAL LETTER E WITH DOT ABOVE
0x0117	LATIN SMALL LETTER E WITH DOT ABOVE
0x0118	LATIN CAPITAL LETTER E WITH OGONEK
0x0119	LATIN SMALL LETTER E WITH OGONEK
0x011A	LATIN CAPITAL LETTER E WITH CARON
0x011B	LATIN SMALL LETTER E WITH CARON
0x011C	LATIN CAPITAL LETTER G WITH CIRCUMFLEX
0x011D	LATIN SMALL LETTER G WITH CIRCUMFLEX
0x011E	LATIN CAPITAL LETTER G WITH BREVE
0x011F	LATIN SMALL LETTER G WITH BREVE
0x0120	LATIN CAPITAL LETTER G WITH DOT ABOVE
0x0121	LATIN SMALL LETTER G WITH DOT ABOVE
0x0122	LATIN CAPITAL LETTER G WITH CEDILLA
0x0123	LATIN SMALL LETTER G WITH CEDILLA
0x0124	LATIN CAPITAL LETTER H WITH CIRCUMFLEX
0x0125	LATIN SMALL LETTER H WITH CIRCUMFLEX
0x0126	LATIN CAPITAL LETTER H WITH STROKE
0x0127	LATIN SMALL LETTER H WITH STROKE
0x0128	LATIN CAPITAL LETTER I WITH TILDE
0x0129	LATIN SMALL LETTER I WITH TILDE
0x012A	LATIN CAPITAL LETTER I WITH MACRON
0x012B	LATIN SMALL LETTER I WITH MACRON
0x012E	LATIN CAPITAL LETTER I WITH OGONEK
0x012F	LATIN SMALL LETTER I WITH OGONEK
0x0130	LATIN CAPITAL LETTER I WITH DOT ABOVE
0x0131	LATIN SMALL LETTER DOTLESS I
0x0132	LATIN CAPITAL LIGATURE IJ
0x0133	LATIN SMALL LIGATURE IJ
0x0134	LATIN CAPITAL LETTER J WITH CIRCUMFLEX
0x0135	LATIN SMALL LETTER J WITH CIRCUMFLEX
0x0136	LATIN CAPITAL LETTER K WITH CEDILLA
0x0137	LATIN SMALL LETTER K WITH CEDILLA
0x0138	LATIN SMALL LETTER KRA
0x0139	LATIN CAPITAL LETTER L WITH ACUTE
0x013A	LATIN SMALL LETTER L WITH ACUTE
0x013B	LATIN CAPITAL LETTER L WITH CEDILLA
0x013C	LATIN SMALL LETTER L WITH CEDILLA
0x013D	LATIN CAPITAL LETTER L WITH CARON
0x013E	LATIN SMALL LETTER L WITH CARON

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 373 of 861

Code	ISO/IEC 10646-1 [15] Character name
0x013F	LATIN CAPITAL LETTER L WITH MIDDLE DOT
0x0140	LATIN SMALL LETTER L WITH MIDDLE DOT
0x0141	LATIN CAPITAL LETTER L WITH STROKE
0x0142	LATIN SMALL LETTER L WITH STROKE
0x0143	LATIN CAPITAL LETTER N WITH ACUTE
0x0144	LATIN SMALL LETTER N WITH ACUTE
0x0145	LATIN CAPITAL LETTER N WITH CEDILLA
0x0146	LATIN SMALL LETTER N WITH CEDILLA
0x0147	LATIN CAPITAL LETTER N WITH CARON
0x0148	LATIN SMALL LETTER N WITH CARON
0x014A	LATIN CAPITAL LETTER ENG
0x014B	LATIN SMALL LETTER ENG
0x014C	LATIN CAPITAL LETTER O WITH MACRON
0x014D	LATIN SMALL LETTER O WITH MACRON
0x0152	LATIN CAPITAL LIGATURE OE
0x0153	LATIN SMALL LIGATURE OE
0x0154	LATIN CAPITAL LETTER R WITH ACUTE
0x0155	LATIN SMALL LETTER R WITH ACUTE
0x0156	LATIN CAPITAL LETTER R WITH CEDILLA
0x0157	LATIN SMALL LETTER R WITH CEDILLA
0x0158	LATIN CAPITAL LETTER R WITH CARON
0x0159	LATIN SMALL LETTER R WITH CARON
0x015A	LATIN CAPITAL LETTER S WITH ACUTE
0x015B	LATIN SMALL LETTER S WITH ACUTE
0x015C	LATIN CAPITAL LETTER S WITH CIRCUMFLEX
0x015D	LATIN SMALL LETTER S WITH CIRCUMFLEX
0x015E	LATIN CAPITAL LETTER S WITH CEDILLA
0x015F	LATIN SMALL LETTER S WITH CEDILLA
0x0160	LATIN CAPITAL LETTER S WITH CARON
0x0161	LATIN SMALL LETTER S WITH CARON
0x0162	LATIN CAPITAL LETTER T WITH CEDILLA
0x0163	LATIN SMALL LETTER T WITH CEDILLA
0x0164	LATIN CAPITAL LETTER T WITH CARON
0x0165	LATIN SMALL LETTER T WITH CARON
0x0166	LATIN CAPITAL LETTER T WITH STROKE
0x0167	LATIN SMALL LETTER T WITH STROKE
0x0168	LATIN CAPITAL LETTER U WITH TILDE
0x0169	LATIN SMALL LETTER U WITH TILDE
0x016A	LATIN CAPITAL LETTER U WITH MACRON
0x016B	LATIN SMALL LETTER U WITH MACRON
0x016C	LATIN CAPITAL LETTER U WITH BREVE
0x016D	LATIN SMALL LETTER U WITH BREVE
0x016E	LATIN CAPITAL LETTER U WITH RING ABOVE
0x016F	LATIN SMALL LETTER U WITH RING ABOVE
0x0172	LATIN CAPITAL LETTER U WITH OGONEK
0x0173	LATIN SMALL LETTER U WITH OGONEK
0x0174	LATIN CAPITAL LETTER W WITH CIRCUMFLEX
0x0175	LATIN SMALL LETTER W WITH CIRCUMFLEX
0x0176	LATIN CAPITAL LETTER Y WITH CIRCUMFLEX
0x0177	LATIN SMALL LETTER Y WITH CIRCUMFLEX
0x0178	LATIN CAPITAL LETTER Y WITH DIAERESIS
0x0179	LATIN CAPITAL LETTER Z WITH ACUTE
0x017A	LATIN SMALL LETTER Z WITH ACUTE
0x017B	LATIN CAPITAL LETTER Z WITH DOT ABOVE
0x017C	LATIN SMALL LETTER Z WITH DOT ABOVE
0x017D	LATIN CAPITAL LETTER Z WITH CARON
0x017E	LATIN SMALL LETTER Z WITH CARON
0x01CD	LATIN CAPITAL LETTER A WITH CARON
0x01CE	LATIN SMALL LETTER A WITH CARON
0x02C6	MODIFIER LETTER CIRCUMFLEX ACCENT
0x02C7	CARON (Mandarin Chinese third tone)
0x02C9	MODIFIER LETTER MACRON (Mandarin Chinese first tone)
0x02D8	BREVE
0x02D9	DOT ABOVE (Mandarin Chinese light tone)

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 374 of 861

Code	ISO/IEC 10646-1 [15] Character name
0x02DA	RING ABOVE
0x02DB	OGONEK
0x02DC	SMALL TILDE
0x066B	ARABIC DECIMAL SEPARATOR
0x1E80	LATIN CAPITAL LETTER W WITH GRAVE
0x1E81	LATIN SMALL LETTER W WITH GRAVE
0x1E82	LATIN CAPITAL LETTER W WITH ACUTE
0x1E83	LATIN SMALL LETTER W WITH ACUTE
0x1E84	LATIN CAPITAL LETTER W WITH DIAERESIS
0x1E85	LATIN SMALL LETTER W WITH DIAERESIS
0x1EF2	LATIN CAPITAL LETTER Y WITH GRAVE
0x1EF3	LATIN SMALL LETTER Y WITH GRAVE
0x2007	FIGURE SPACE
0x2013	EN DASH
0x2014	EM DASH
0x2018	LEFT SINGLE QUOTATION MARK
0x2019	RIGHT SINGLE QUOTATION MARK
0x201A	SINGLE LOW-9 QUOTATION MARK
0x201C	LEFT DOUBLE QUOTATION MARK
0x201D	RIGHT DOUBLE QUOTATION MARK
0x201E	DOUBLE LOW-9 QUOTATION MARK
0x2022	BULLET
0x2026	HORIZONTAL ELLIPSIS
0x2030	PER MILLE SIGN
0x2039	SINGLE LEFT-POINTING ANGLE QUOTATION MARK
0x203A	SINGLE RIGHT-POINTING ANGLE QUOTATION MARK
0x2044	FRACTION SLASH
0x20AC	EURO SIGN
0x2122	TRADE MARK SIGN
0x2190	LEFTWARDS ARROW
0x2191	UPWARDS ARROW
0x2192	RIGHTWARDS ARROW
0x2193	DOWNWARDS ARROW
0x2212	MINUS SIGN
0x2214	DOT PLUS
0x2215	DIVISION SLASH
0x221E	INFINITY
0x266B	BEAMED EIGHTH NOTES
0x2713	CHECK MARK
0x2717	BALLOT X

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 375 of 861

Annex F (informative): Authoring and implementation guidelines

F.1 Authoring Guidelines

- Authoring guidelines are needed to specify those methods, classes and interfaces which are intended for use by implementations of JMF players. These methods, classes and interfaces are not intended for use by applications except for this purpose and that should be made clear.
- Authoring guidelines are needed to make it clear that it is optional for JMF controls to have an associated java.awt component. This is in the JMF documentation but the phrasing implies this an exceptional case. In many GEM receivers, the presence of such a component would be the exceptional case.To be completed.

F.2 Implementation Guidelines

To be completed.

F.3 Authoring guidelines for DVB-J

To be completed.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Annex G (normative): Minimum platform capabilities

G.1 Graphics

In the area of graphics capability the following requirements are made on GEM terminals.

G.1.1 Device capabilities

G.1.1.1 General

- The number of DVB-J applications concurrently owning instances of `HScene` is not limited except by underlying platform resources like the total memory of the GEM terminal. The GEM terminal is required to support either the "Platforms Supporting a Restricted Multi-Window System" or the "Platforms Supporting a Full Multi-Window System" implementation scenarios as defined in `org.havi.ui.HSceneFactory`. The implementation scenario "Platforms Supporting a Single Window System" is not a valid choice for GEM.
- The GEM terminal is not required to provide a mechanism for the end-user to change user input focus between `HScenes`. Hence GEM applications wishing to receive user input focus must explicitly request it.
- The GEM terminal shall implement at least one `HGraphicsDevice` which shall be full screen.
- The GEM terminal shall implement at least one `HBackgroundDevice`. These are always full screen.
- The GEM terminal shall implement at least one `HVideoDevice` which is always capable of being configured to be full screen.
- The GEM terminal shall implement at least one `HScreen` which shall support at least one video, graphics and background device as defined immediately above.

G.1.1.2 Standard definition

The following tables define HScreenConfigurations and sets of coherent HScreenConfigurations for GEM terminals not supporting high definition video.

GEM terminals capable of 4:3 output and 16:9 output (such as set top boxes but also IDTVs capable of a "reduced scan") shall, regardless of the aspect ratio of the attached display (to the extent this is known), support all mandatory configurations and sets of coherent screen configurations in the appropriate table. GEM terminals capable of outputting a signal in only one aspect ratio shall support all mandatory configurations and sets of coherent configurations for that screen aspect ratio.

Table 82: Sets of coherent configurations for standard definition - 625-line markets

Set	Screen Aspect Ratio	HGraphics Configuration		HVideo Configuration		HBackground Configuration		Notes
		Resolution	Pixel aspect ratio	Resolution	Pixel aspect ratio	Resolution	Pixel aspect ratio	
1,2	4:3	720x576	56/45	720x576	16/15	720x576	16/15	Default for 4:3 output
3,4	4:3	720x576	16/15	720x576	16/15	720x576	16/15	
5,6	*4:3*	*768x576*	*1/1*	720x576	16/15	720x576	16/15	
7,8	16:9	720x576	56/45	720x576	64/45	720x576	64/45	Default for 16:9 output
9,10	16:9	720x576	64/45	720x576	64/45	720x576	64/45	
11,12	*16:9*	*1024x576*	*1/1*	720x576	64/45	720x576	64/45	

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 83: Sets of coherent configurations for standard definition - 525-line markets

Set	Screen Aspect Ratio	HGraphics Configuration		HVideo Configuration		HBackground Configuration		Notes
		Resolution	Pixel aspect ratio	Resolution	Pixel aspect ratio	Resolution	Pixel aspect ratio	
1,2	4:3	720x480	47/45	720x480	8/9	720x480	8/9	Default for 4:3 output
3,4	4:3	720x480	8/9	720x480	8/9	720x480	8/9	
5,6	*4:3*	*640x480*	*1/1*	720x480	8/9	720x480	8/9	
7,8	16:9	720x480	47/45	720x480	32/27	720x480	32/27	Default for 16:9 output
9,10	16:9	720x480	32/27	720x480	32/27	720x480	32/27	
11,12	*16:9*	*854x480*	*1/1*	720x480	32/27	720x480	32/27	

The following additional requirements shall apply concerning both the above tables:

- HScreenConfigurations and sets of coherent screen configurations shown in *italics* are always optional.
- Each row shall correspond to two sets of coherent configurations since for each possible combination of a HGraphicsConfiguration and a HVideoConfiguration, 2 HBackgroundConfigurations shall be supported - one a HStillImageBackgroundConfiguration (supporting the display of MPEG stills) and one just a HBackgroundConfiguration only supporting a single colour.
- All configurations in the table shall exactly cover the full area from (0,0) to (1,1).

NOTE 1: For video, whether the video picture covers the entire HVideoDevice depends on factors including scaling, positioning and decoder format conversion.

- Where a GEM application has the HGraphicsDevice reserved, the receiver's picture shape and/or wide screen signalling shall be set to the output aspect ratio of the HGraphicsDevice. Where the HGraphicsDevice is not reserved by a GEM application, the picture shape and/or wide screen signalling shall be set according to the configuration of the HVideoDevice and the decoder format conversion being applied to the video.
- Each required HVideoConfiguration shall have its pixels aligned (as defined by HScreenConfigTemplate.VIDEO_GRAPHICS_PIXEL_ALIGNED) with all required HGraphicsConfigurations and vice-versa.
- HGraphicsConfigurations whose pixel aspect ratio is 56/45 are HEmulatedGraphicsConfigurations corresponding to the pixel aspect ratio of a logical 14:9 screen as defined in clause 13.3.7 "14:9 Aspect Ratio Support".

NOTE 2: Although default configurations are shown, application developers should note that another application may already have changed the graphics configuration from the default before they start running or may change it while they are running.

G.1.1.3 High definition

GEM terminals supporting HD shall support at least 2 HGraphicsDevices as follows;

- The one in front shall have 4 possible HGraphicsConfigurations:
  - full screen 1 280 x 720;
  - full screen 960 x 540;
  - full screen standard definition (16:9);
  - full screen standard definition (14:9 - HEmulatedGraphicsConfiguration).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

- The one behind shall have 3 possible HGraphicsConfigurations:
  - full screen 1 280 x 720;
  - full screen standard definition (16:9);
  - full screen standard definition (14:9 - HEmulatedGraphicsConfiguration).

GEM terminals shall support the simultaneous display of full screen 1 280 x 720 and full screen standard definition graphics. GEM does not define how this is supported. Some possibilities include actual hardware graphics planes, hardware assisted composition ("blitting") and rendered into a single actual hardware graphics plane (e.g. by applying a transformation in the graphics library). Where a single real hardware graphics plane is used for both these resolutions, it shall have a resolution of at least 1 280 x 720. In this case, it is an allowed implementation option that applications in the rear HGraphicsDevice are clipped by all applications in front of them, both those in the same HGraphicsDevice area and those in the front HGraphicsDevice.

The resolution of 960 x 540 shall always be supported with an actual hardware graphics plane whose resolution is at least 960 x 540. GEM permits (but does not require) that the front HGraphicsDevice can only be configured to 960 x 540 when the rear HGraphicsDevice is disabled as defined by HScreenConfigTemplate.DISABLED. Applications which want to use 960 x 540 should be careful to ask for that resolution using HScreen.getCoherentConfigurations without putting requirements on the configuration of the rear graphics plane.

In the absence of GEM application control, the default shall be full screen 1 280 x 720 graphics in front of full screen SD graphics.

NOTE: Unbound applications running on GEM terminals supporting HD should ensure that a resolution of 1 280 x 720 is always available except when it has been arranged that no applications require it. Conversely service bound GEM applications wishing to obtain a resolution of 960 x 540 on GEM terminals supporting unbound applications may need to co-ordinate this with the owner of the unbound applications.

When HD video is being decoded, the resolution of the HVideoDevice is not specified by GEM. Some possible resolutions include the following;

- The resolution of the video being decoded. Typically used if the HD display does the conversion between the transmitted video resolution and the display resolution.
- The resolution of the HD display - assuming the GEM terminal has this information. Typically used for (old) displays which only support one resolution on their input.
- Some other (neutral) resolution.

In any case, it is expected that there is a single HVideoConfiguration for HD video decoding Hence it is meaningless for GEM applications to reserve or set the HVideoConfiguration when HD video is being decoded.

When HD video is being decoded, clause X.1.6 "MPEG I frame and Video drips" of the present document defines that support for MPEG I-frames (and hence HStillImageBackgroundConfigurations) is optional. Hence HBackgroundDevices shall support at least the following HBackgroundConfigurations;

- An HStillImageBackgroundConfiguration which is required to be available except when HD video is being decoded (in which case it is optional).
- A full screen HBackgroundConfiguration which is not an HStillImageBackgroundConfiguration and which is valid at all times.

The following tables define mandatory and optional HScreenConfigurations and sets of coherent screen configurations for high definition markets. Optional HScreenConfigurations and sets of coherent screen configurations are shown in *italics.*

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 84: Sets of coherent configurations for high definition

Set Number	Video Being Decoded	Front HGraphics Configuration	Rear HGraphics Configuration	HVideo Configuration	HBackground Configuration	Notes
1,2	SD	1280x720	16:9 SD	16:9 SD	16:9 SD, one configuration capable of showing a still and one not.	
3,4	SD		14:9 SD			default for SD
5,6	SD	16:9 SD	1280x720			
7,8	SD	14:9 SD				
9,10	SD	960x540	Not specified in present document			
11,12	*SD*	*1920x1080*				
13	HD	1280x720	14:9 SD	HD video	Not specified in present document	default for HD
14	HD	1280x720	16:9 SD	HD video		
15	HD	16:9 SD	1280x720	HD video		
16	HD	14:9 SD	1280x720	HD video		
17	HD	960x540	Not specified in present document	HD video		
18	*HD*	*1920x1080*		HD video		

The following additional requirements shall apply concerning table 84:

- "16:9 SD" shall mean the appropriate resolution with the terminal in standard definition mode.
- "14:9 SD" shall mean the appropriate resolution with the terminal in standard definition mode.
- Each of the rows shown as corresponding to two sets of coherent configurations means set where the HbackgroundConfiguration shall be a HStillImageBackgroundConfiguration (supporting the display of MPEG stills) and one set where it is just a HBackgroundConfiguration only supporting a single colour.
- "HD video" means the single HVideoConfiguration for HD video referred to earlier in this clause.
- HScreenConfigurations and sets of coherent screen configurations shown in *italics* are always optional.

G.1.2 Video presentation capabilities

- GEM terminal specifications for packaged media targets require support for DFC_PROCESSING_FULL only.
- The following set of standard decoder format conversions shall be supported by all GEM terminals (except packaged media targets):
  - DFC_PROCESSING_CCO.
  - DFC_PROCESSING_FULL.
  - DFC_PROCESSING_LB_16_9.
  - DFC_PROCESSING_PAN_SCAN.
- The following modes are optional:
  - DFC_PROCESSING_LB_14_9.
  - DFC_PROCESSING_LB_2_21_1_ON_16_9.
  - DFC_PROCESSING_LB_2_21_1_ON_4_3.
- GEM terminals are required to support displaying MPEG video with horizontal scale factors of 2, 3/2, 1, 3/4, 1/2, 1/3, 1/4, and vertical scale factors of 2, 1, 1/2, 1/3, 1/4. At least half of the decoded image height must remain within the display raster. For horizontal scaling factors of full size and above, at least one quarter of the decoded image width must remain within the display raster. For horizontal scaling factors below full size, at least half the decoded image width must remain within the display raster. Use of the scale factor 0 for both vertical and horizontal scaling shall result in the video being discarded and the video plane being transparent to the plane behind it, as defined by the GEM graphics reference model. These requirements also apply to HD video on GEM terminals supporting that feature.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

NOTE: GEM does not specify behaviour when either but not both of the vertical or the horizontal scaling factors are zero.

- Support for component-based JMF players is only required in GEM terminals that:

  a) Support the simultaneous decoding of more than one broadcast video stream (as defined in clause 7.2.2, "Video") on a single display (HScreen instance).

  b) Expose through the GEM-defined API, the ability to initiate and/or control such decoding.

  In these GEM terminals, all JMF players shall be able to be used as a component-based player when decoding video. In other GEM terminals, this support is optional.

G.1.3 Image processing capabilities

- All DVBGraphics objects shall support SRC and CLEAR and SRC_OVER.
  When SRC_OVER is used with DVBGraphics objects with a sample model of the type TYPE_BASE a perfect result is only guaranteed to be produced with alpha values of 0 and 1. Alpha values other than 0 and 1 can be approximated. DVBGraphics object with a type of TYPE_ADVANCED will produce a result as expected but those SRC_OVER operations are likely to be slow.

- DVBGraphics object created from a DVBBufferedImage with the type TYPE_ADVANCED shall perform SRC_OVER operations without approximations of the compositing rule.

G.1.3.1 Composition rules

GEM terminals are required to implement at least the SRC, SRC_OVER and CLEAR rules when composing graphics over video and graphics over graphics.

G.1.4 Alpha capabilities

In the composition of the graphics (AWT/HAVi components) with background and video planes (see figure 23), GEM terminals are required to implement at least 16 levels of alpha evenly spaced between 0 (completely transparent) and 255 (opaque).

Allowed approximations for the composition of graphics over other graphics are defined in clause 13.6.1.1.2 "Graphics over other graphics".

G.1.5 Colour capabilities

The colour model is a "true colour" one. There shall be at least 4 bits per pixel for each of R. G and B.

G.1.6 MPEG I frame and Video drips

The minimum positioning and scaling capabilities defined above for MPEG video shall also apply to MPEG I frame and Video drips.

The GEM terminal shall support at least one `HStillImageBackgroundConfiguration`.

GEM terminals shall support simultaneously the display of already decoded MPEG I-frames and the decoding and display of video. It is implementation dependent as to whether the decoding of MPEG I-frames is supported simultaneously with the decoding of video.

The display of I-frames simultaneously with HD video is optional.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

G.2 Audio

No audio mixing is required.

Audio played from memory may pre-empt any audio from the transport stream. This may disturb decoding of any broadcast video stream.

Audio from memory shall be output in preference to audio from stream if overall audio output has not been disabled by the user. On platforms capable of mixing audio from memory with audio from the stream it shall do this if there is a stream playing. Where audio from the stream is interrupted, decoding of it shall automatically resume when audio from memory ceases if the stream concerned is still playing. This only applies where the audio from memory and the stream are both under the control of the same GEM application. Where multiple applications are involved see clause 9.4 "Inter application resource management".

G.3 Video

The GEM terminal is only required to support decoding of a single video stream at a given time. The number of implemented video decoders will affect the functionality of the video and background devices.

GEM terminals shall be able to present un-synchronised broadcast video and audio where explicitly requested by GEM applications. See clause 11.4.2.5.5, "Clarifications".

G.4 Resident fonts and text rendering

G.4.1 The built-in font

At least the font Tiresias [63] shall be provided.

NOTE 1: Tiresias is the trade name of a product supplied by Bitstream and owned by the Royal National Institute of the Blind. This information is given for the convenience of users of GEM and does not constitute an endorsement by ETSI of the product named. Equivalent products may be used if they can be shown to lead to the same results.

NOTE 2: As described in clause 7.3, "Resident fonts" in the present document, the resident font might not be built into the terminal in all cases.

The built-in font shall be supported with the weight ("PLAIN"). The default size shall be 26 points.

The built-in font shall be rendered from an outline representation with a continuous range of point sizes from 12 to 96.

NOTE 3: Application developers should be aware that implementations may cache glyphs for recently used sizes. Using many different sizes simultaneously may degrade performance.

Table 85 gives example sizes for various use cases.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 85: Example font sizes and use cases

Size (points)	TV lines over "Cap-V" (see notes 1 and 2)	Informative Name
36	24	Heading / Large subtitle
31	21	Subtitle
26	18	Body
24	16	Footnote
NOTE 1: The primary definition of the character size is the font size in points, the height of a capital letter "V" in TV lines is provided for information only.		
NOTE 2: The second column of this table is informative and it may not apply to all GEM terminal specifications. For clarity GEM terminal specifications may specify an equivalent of these values, though if they do not this does not imply that the second column of the table is accurate for these specifications.		

G.4.2 Presentation to DVB-J

The embedded font "Tiresias" shall have:

- The logical name "SansSerif" (for example returned by `java.awt.Toolkit.getFontList`).
- The family name "Tiresias" (for example returned by `java.awt.Font.getFamily`).
- The font face name "Tiresias PLAIN".

G.4.3 Text directions

The `DVBTextLayoutManager` is only required to support the following configuration of text direction:

- LINE_ORIENTATION_HORIZONTAL and START_CORNER_UPPER_LEFT.

G.5 Input events

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

Some downloaded resident applications specified as extensions to GEM may perform some of the functions of the GEM navigator, e.g. the monitor application defined OCAP [5]. In this case, such downloaded software shall implement a policy that is consistent with the requirements of GEM.

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

Table 86: Minimum set of input events

Input event
VK_0 to VK_9
VK_UP
VK_DOWN
VK_LEFT
VK_RIGHT
VK_ENTER
VK_TELETEXT (optional)
VK_COLORED_KEY_0
VK_COLORED_KEY_1
VK_COLORED_KEY_2
VK_COLORED_KEY_3 (optional)

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 87: Additional input events required by privileged applications

Input event
VK_CHANNEL_DOWN
VK_CHANNEL_UP
VK_HOME

These tables define the minimum set of input events which shall be available to the set of running GEM applications if they are interested in receiving them. For DVB-J applications, this is described in more detail in clause 11.4.1.4, "Television viewing mode".

NOTE 1: They are not guaranteed to be always available to any one GEM application because another application running at the same time may have one of these events exclusively reserved. The application with focus (if any) always receives all of these events unless another application within the same service has requested and been granted exclusive access to one or more events. The process for event distribution for DVB-J applications is described in more detail in annex J "DVB-J event API".

NOTE 2: The user input device for a GEM terminal may support more events than this however this is implementation dependent. If more events than this are supported, it is equally implementation dependent whether the additional events are sent to GEM applications or sent to the GEM navigator. Events which are always sent to the GEM navigator may not be visible at all to GEM applications. For example, a GEM receiver using a conventional remote control will probably have program up/program down keys which are only ever sent to the navigator and cause service selection when received there.

NOTE 3: The specification for `java.awt.event.keyEvent`, `org.havi.ui.event.HRcEvent` and `org.dvb.event.UserEvent` do not define which VK_ codes have "valid" Unicode characters. Hence DVB-J applications cannot rely on the return value of the `getKeyChar` method for KEY_PRESSED or KEY_RELEASED events. Applications which want to use Unicode characters should use KEY_TYPED events instead.

GEM terminals intending to provide a remote control key (or equivalent) labelled "back" (or equivalent) shall use VK_F9 as its code. GEM applications decoding MHEG-5 Broadcast Profile content shall use this key as the MHEG-5 "cancel key function". The present document does not require support for this key in all GEM terminals.

The `HRcCapabilities.getRepresentation(int)` method shall return an `HEventRepresentation` instance for each input event in the above table. Each such `HEventRepresentation` instance shall return at least one of ER_TYPE_COLOR, ER_TYPE_STRING or ER_TYPE_SYMBOL from calls to its `getType()` method. Where ER_TYPE_SYMBOL is returned, the corresponding `java.awt.Image` instance shall have the size of 32 by 32. The representations supported shall match the user input device generating the input events listed above which is shipped with the GEM terminal.

NOTE 4: Application developers need to be aware that a small proportion of end-users will be using user input devices not shipped with the GEM terminal, e.g. replacements and/or programmable remote controls. There is no requirement for GEM terminals to detect this.

NOTE 5: This requires a meaningful representation of how a given supported keycode is generated.

The following mapping from key codes to remote control labelling is recommended:

- VK_COLORED_KEY_0 Red.
- VK_COLORED_KEY_1 Green.
- VK_COLORED_KEY_2 Yellow.
- VK_COLORED_KEY_3 Blue.

Additionally, it is recommended that these keys be in the order red, green yellow then blue.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

G.6 Memory

In order to be able to execute GEM conformance tests, the following minimum memory requirements are defined for GEM terminals. All of these are to be measured during normal usage and operational conditions of a GEM terminal. All are to be measured in the `initXlet` method of a DVB-J application which is both the only auto-start application signalled in a service and the only application running at that time.

- Enough memory to successfully load any arbitrary 262 144 (or less) Bytes of Java class files into the memory space of the Java virtual machine. Execution of code called as part of initializing fields in classes is excluded from consideration as part of "load"ing here. RAM usage by the bytecode verifier is included in consideration as part of "load"ing here.

Enough memory to do the above and individually each of the following:

- Enough memory to successfully create a Java byte array of lengths from 1 entry to 262 144 entries.
- Enough memory to successfully load a PNG image whose size is the same as the resolution of an `HGraphicsDevice` at standard definition, as discussed in clause G.1.
- Enough memory to successfully load from file and play from memory 5 seconds of audio at 128 kbit/s (where kbit/s is as used in TS 101 154 [2]). It shall be measured using files that do not include any optional extension fields.
- Enough memory to successfully allocate an array of `java.lang.Object` of length 16 384 and fill each element of this array with a distinct instance of `java.lang.Object`.

The memory requirements detailed in this clause are not exhaustive. For example, the specific requirement concerning an array of type byte in no way implies that GEM terminals are exempt from requirements found elsewhere in the present document (including normatively referenced specifications) for supporting arrays of other types.

NOTE: Additional detail may be added to these requirements in order to properly enable the GEM conformance tests.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

G.7 Other resources

Table 88: Minimum requirements for other resources

Feature	Specification
gamma correction in the receiver	none
HAVi mattes	Platforms are not required to implement the functionality of mattes in HAVi. Non-implementation should be implemented as specified by HAVi.
Overlapping applications	GEM terminals are not required to support overlapping top level UI containers (e.g. HScenes where DVB-J applications are concerned).
AIT section filtering	The implementation is not required to dedicate more than one section filter to monitoring the AIT. NOTE 1: This entry applies only to GEM terminal specifications that include the functional equivalent named "Application Signalling" as defined in clause 15.6, "Functional equivalents". However, attention is drawn to the requirement on detecting Application Description changes in clause 10.4.1.
Key lengths for TLS	Receivers shall support certificate key lengths up to and including 2 048 bits for TLS (see clause 12.10, "Security on the return channel"). NOTE 2: The key lengths for application authentication apply only to codesigning using the MHP model.
Key lengths for Application Authentication	Receivers shall support certificate key lengths up to and including 2 048 bits for application authentication (see clause 12.2, "Authentication of applications").
Internet clients	WWW and email clients required, usenet news client optional.
Non-CA Smart Card Reader	GEM terminals are not required to include this
Stored services	No requirement for persistent storage for stored services even if other forms of persistent storage are supported.
PSTN/ISDN modem (where present)	Where such a modem is present, the GEM terminal should provide a mechanism to configure a dialing prefix, e.g. '0', to be dialled in front of the number specified by the GEM application. Where such a prefix is supported, this shall be invisible to GEM applications, for example, it shall not need to be included in entries in the permission request file.

NOTE: The values in the table below are set for the purposes of conformance testing and should not be used by application or GEM terminal developers as being indicative of the capabilities of commercial products.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Table 89: Minimum requirements for other resources for conformance purposes

Feature	Specification
Application accessible timers (see note 2)	At least 4 timers for each ServiceContext which can be presenting GEM applications at the same time. (i.e. shared between the applications signalled as part of the same service) (see note 1).
MPEG-2 transport stream network interface	Shall support at least one network interface enabling reception of an MPEG-2 transport stream and selection of that transport stream from among those available to be received. This shall support those broadcast channel protocols supported by the GEM terminal (see clause 6.2, "Broadcast channel protocols") and the GEM APIs defined to interface to these protocols and to control these interfaces.
Bidirectional IP network interface	GEM terminals supporting the interactive broadcast profile shall support at least one network interface for bidirectional IP traffic. This shall support those interaction channel protocols supported by the GEM terminal (see clause 6.3, "Interaction channel protocols") and the GEM APIs defined to interface to these protocols and to control these interfaces.
Conditional access	None required. The absence or optional presence of one shall be correctly reported through the Conditional access API. Local regulation may require more support than this minimum. See note 4.
Persistent storage	At least 131 072 bytes. This shall not include the requirements of clause 12.12, "Platform minima" for persistent storage of CRLs and RCMMs.
Application accessible MPEG-2 section filters (see note 2)	At least 2 shared among all applications signalled as part of the same service (see note 1).
Application accessible DVB-J threads (see note 2)	At least 4 shared among all applications signalled as part of the same service. Threads created by the platform and used to call methods of the application are excluded from this number (see note 1).
Applications in a service	GEM terminals shall not impose any arbitrary limit on the number of applications which they can support at the same time.
Memory for stored services as defined in clause 9.6.2, "Stored services"	Either 0 or greater than or equal to 65 536 bytes. This memory shall be non-volatile. If greater than zero, there shall be a means to empty this memory, at least for the purposes of running conformance tests (see note 3).
NOTE 1: These requirements apply to one set of GEM applications signalled as part of the same service. If an GEM terminal supports simultaneous execution of more than one set of signalled applications then it shall make available at least these minimum resources for each set of signalled applications which can be executed simultaneously. NOTE 2: "Application accessible" means guaranteed to be accessible to GEM applications through the API defined in the present document for the feature concerned. This must be regardless of the extent of any usage of that feature or function as part of the GEM terminal implementation. NOTE 3: Implementors should choose an actual size based on the sizes of GEM applications found in their market and not based on this largely arbitrary number. NOTE 4: These requirements apply only to GEM terminal specifications that support the functional equivalent named "Conditional Access".	

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 387 of 861

Annex H (normative): Extensions

Private protocols and possibly APIs are not precluded and are outside of the scope of the GEM specification.

The addition of public or protected constructors, methods or fields to classes and interfaces in the `org.dvb` namespace is not allowed.

Restrictions on proprietary extensions to the `org.havi.ui` packages are found in section 7.2.2 ("Profile #1: DCMs and Application Modules") of HAVi [50].

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Annex I (normative): DVB-J fundamental classes

Package

org.dvb.lang

Description

Provides those core platform related features not found in the java.lang package.

Class Summary	
Classes DVBClassLoader	This class loader is used to load classes and resources from a search path of URLs referring to locations where Java class files may be stored.

org.dvb.lang

DVBClassLoader

Declaration

```
public abstract class DVBClassLoader extends java.security.SecureClassLoader
```

```
java.lang.Object
  |
  +--java.lang.ClassLoader
        |
        +--java.security.SecureClassLoader
              |
              +--org.dvb.lang.DVBClassLoader
```

Description

This class loader is used to load classes and resources from a search path of URLs referring to locations where Java class files may be stored.

The classes that are loaded are by default only allowed to load code through the parent classloader, or from the URLs specified when the DVBClassLoader was created.

Constructors

DVBClassLoader(URL[])

```
public DVBClassLoader(java.net.URL[] urls)
```

Constructs a new DVBClassLoader for the given URLs. The URLs will be searched in the order specified for classes and resources.

If there is a security manager, this method first calls the security manager's `checkCreateClassLoader` method to ensure creation of a class loader is allowed.

Parameters:

`urls` - the URLs from which to load classes and resources.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Throws:

`java.lang.SecurityException` - if a security manager exists and its `checkCreateClassLoader` method does not allow creation of a class loader.

See Also:

`java.lang.SecurityManager.checkCreateClassLoader()`

DVBClassLoader(URL[], ClassLoader)

```
public DVBClassLoader(java.net.URL[] urls, java.lang.ClassLoader parent)
```

Constructs a new DVBClassLoader for the given URLs. The URLs will be searched in the order specified for classes and resources.

If there is a security manager, this method first calls the security manager's `checkCreateClassLoader` method to ensure creation of a class loader is allowed.

Parameters:

`urls` - the URLs from which to load classes and resources

`parent` - the parent classloader for delegation

Throws:

`java.lang.SecurityException` - if a security manager exists and its `checkCreateClassLoader` method does not allow creation of a class loader.

See Also:

`java.lang.SecurityManager.checkCreateClassLoader()`

Methods

findClass(String)

```
protected java.lang.Class findClass(java.lang.String name)
    throws ClassNotFoundException
```

Finds and loads the class with the specified name from the URL search path. Any URLs are searched until the class is found.

Overrides:

`findClass` in class `ClassLoader`

Parameters:

`name` - the name of the class.

Returns:

the resulting class.

Throws:

`java.lang.ClassNotFoundException` - if the named class could not be found.

newInstance(URL[])

```
public static org.dvb.lang.DVBClassLoader newInstance(java.net.URL[] urls)
```

Creates a new instance of DVBClassLoader for the specified URLs. If a security manager is installed, the `loadClass` method of the DVBClassLoader returned by this method will invoke the `SecurityManager.checkPackageAccess` method before loading the class.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Parameters:

`urls` - the URLs to search for classes and resources.

Returns:

the resulting class loader

newInstance(URL[], ClassLoader)

```
public static org.dvb.lang.DVBClassLoader newInstance(java.net.URL[] urls,
java.lang.ClassLoader parent)
```

Creates a new instance of DVBClassLoader for the specified URLs. If a security manager is installed, the `loadClass` method of the DVBClassLoader returned by this method will invoke the `SecurityManager.checkPackageAccess` method before loading the class.

Parameters:

`urls` - the URLs to search for classes and resources.

`parent` - the parent class loader for delegation.

Returns:

the resulting class loader.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 391 of 861

Annex J (normative): DVB-J event API

Applications can use the `org.dvb.event` API, to receive events without being focused and/or to have exclusive access to events.

NOTE: The return value of `org.dvb.event.UserEvent.getKeyChar()` is defined in terms of `CHAR_UNDEFINED`. The value of `getKeyChar()` for this method is not defined in an interoperable way. In no case should an interoperable application rely on the return value of `getKeyChar()`; further, the value returned for an undefined key is not defined interoperably and may vary across platform implementations.

J.1 Overview

This API provides a mechanism allowing GEM applications to influence the routing of events to either GEM applications or the navigator. This typically would be used in a mode where the receiver is primarily used for TV viewing, allowing some events to be received by the application, and allowing the navigator to receive those events that are not requested by any GEM application. This API enables GEM applications to choose between the following mechanisms for receiving events:

- through the standard `java.awt` mechanism;
- through the standard `java.awt` mechanism, but for some events to be exclusively accessed by the application;
- through a mechanism defined by this API;
- through the mechanism defined by this API, but for some events to be exclusively accessed by the application.

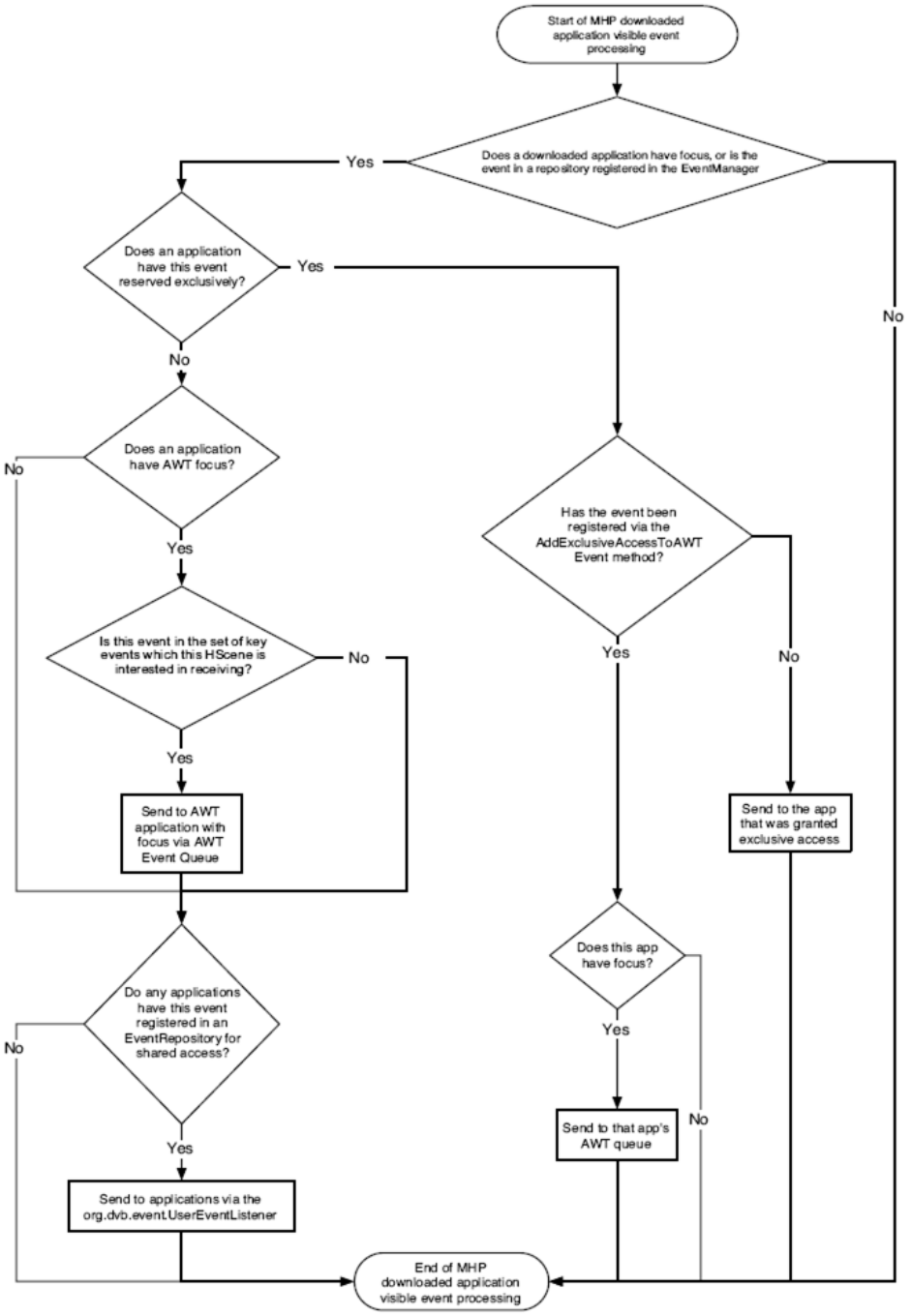
The last two solutions could be used by non-graphical applications in order to receive events that are coming from the user. It could also be used by an invisible application if it wants to be presented when a specific key is pressed.

If an application wants to have exclusive access to some events and to manage them through the java.awt then it must use this API so that it can be aware of the fact that it has lost or gained access to these events. One must notice that an application based on awt event mechanism will receive events only if it is focused.

If an application has acquired exclusive access to receive an event through either of these mechanisms, it will not receive this same event through the other mechanism. An application can obtain exclusive access to an event through only one of the two mechanisms at a time.

The diagram below shows how the GEM terminal decides on the processing of an event (defined by a set of keychar/keycode, event id and modifiers). This includes deciding on the mechanisms by which the event will be dispatched and the appropriate Java class to encapsulate the event for each of these mechanisms.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238



NOTE: In this figure the terms "AWT queue" and "AWT Event Queue" are intended to apply to both events delivered directly through `java.awt` and to events delivered in HAVi event classes. Events delivered in HAVi event classes include `HFocusEvents` with a keycode as the `transfer-id` as well as `HKeyEvents`. Hence if a keycode is exclusively reserved by a `UserEventListener`, or by another application through `EventManager.addExclusiveAccessToAWTEvent()`, this does block `HFocusEvents` being generated if that keycode would be the transfer-id of the `HFocusEvent`.

Figure 47: The GEM downloaded application visible event distribution mechanism

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

J.2 The resource management

An application asking for exclusive access to some events will use the resource framework defined in DAVIC 1.4.1p9 [17] so that it can be aware of the fact that it has lost access to the user events it asked for (see the example below).

J.3 The Event Repository

The UserEventRepository is the class that is used by the application to define the user events it intends to use. For the moment user events are just key events but it is a place-holder for new families of events (voice command for example). If an application asks for an exclusive access to events by means of a repository, this exclusive access will be lost at the time when one of the event is grabbed by another application. User events that can be accessed by an application are defined in the UserEvent class.

J.3.1 Example

```
import org.davic.resources.ResourceClient.* ;
   import org.dvb.event.* ;
   class Example implements UserEventListener, ResourceStatusListener, ResourceClient {
      private int myStatus ;
      public Example () {
         EventManager em ;
         UserEventRepository repository ;
         em = EventManager.getInstance () ;
         repository = new UserEventRepository ("R1") ;

         repository.addKey (UserEvent.VK_ENTER) ;
         em.addUserListener ((UserEventListener)this, (ResourceClient)this, repository) ;
         em.addResourceStatusEventListener (this) ;
      }
      /**
      * methods defined by the UserEventListener interface.
      */
      public void UserEventReceived (UserEvent e) {
      }
      /**
      * Methods defined by the ResourceClient interface.
      */
      /**
      * In the case a cooperative application asks for an user event
      * exclusively used by me.
      */
      public boolean requestRelease(ResourceProxy proxy, Object requestData) {
         String name ;
         // let's retrieve the name of the repository, that I have created, and
         // which contains the user event that the other application asks for.
         name = (RepositoryDescriptor)proxy.getName () ;
         if ((name.compareTo ("R1") == 0) & (myStatus ==  )) {
            // Ok I release this event.
            return true ;
         } else {
            // No I need this event, sorry !
            return false ;
         }
      }
      public void release (ResourceProxy proxy) {

      }
      public void notifyRelease (ResourceProxy proxy) {

      }
      public void statusChanged (ResourceStatusEvent event) {

      }
   }
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

J.4 Unicode

References to "Unicode" in the following API description shall be interpreted as references to ISO/IEC 10646-1 [15].

J.5 Virtual keyboards

On platforms where key events are generated from a sequence of other (intermediate) key events, the intermediate key events shall not be visible to GEM applications by any mechanism. Examples of these intermediate key events include;

- For a virtual keyboard, the sequence of keys used to navigate around that keyboard (e.g. `VK_UP`, `VK_LEFT`, `VK_ENTER`).
- For multi-key press entry (as used in some mobile phones), the keys pressed before the final value is resolved.

Package org.dvb.event

Description

Provides access to user input events before they are processed through the event mechanism of the java.awt package.

The algorithm used for generating UserEvents by the GEM terminal when reporting user input to GEM applications shall be the same as that used for java.awt.event.KeyEvent. For example, pressing the Shift key will cause a KEY_PRESSED event with a VK_SHIFT keyCode, while pressing the 'a' key will result in a VK_A keyCode. After the 'a' key is released, a KEY_RELEASED event will be fired with VK_A, followed by a KEY_TYPED event with a key-Char value of 'A'.

Class Summary	
Interfaces	
`UserEventListener`	The listener interface for receiving user inputs.
Classes	
`EventManager`	The event manager allows an application to receive events coming from the user.
`OverallRepository`	This class defines a repository which initially contains all the user events which can be delivered to an application.
`RepositoryDescriptor`	An instance of this class will be sent to clients of the DVB event API to notify them (through the interface org.davic.resources.ResourceClient) when they are about to lose, or have lost, access to an event source.
`UserEvent`	Represents a user event.
`UserEventAvailableEvent`	This event is sent to the resource status event listeners when user input events which had been exclusively reserved by an application are no longer exclusively reserved.
`UserEventRepository`	The application will use this class to define the events that it wants to receive.
`UserEventUnavailableEvent`	This event is sent to the resource status event listeners when user input events are exclusively reserved by an application.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

org.dvb.event

EventManager

Declaration

```
public class EventManager implements org.davic.resources.ResourceServer

java.lang.Object
  |
  +--org.dvb.event.EventManager
```

All Implemented Interfaces:

```
org.davic.resources.ResourceServer
```

Description

The event manager allows an application to receive events coming from the user. These events can be sent exclusively to an application or can be shared between applications. The Event Manager allows also the application to ask for exclusive access to some events, these events being received either from the standard java.awt event mechanism or by the mechanism defined in this package. The EventManager is either a singleton for each GEM application or a singleton for the GEM terminal.

The right to receive events is considered as the same resource regardless of whether it is being handled exclusively or shared. An application successfully obtaining exclusive access to an event results in all other applications losing access to that event, whether the access of those applications was shared or exclusive.

Constructors

EventManager()

```
protected EventManager()
```

Constructor for instances of this class. This constructor is provided for the use of implementations and specifications which extend the present document. Applications shall not define sub-classes of this class. Implementations are not required to behave correctly if any such application defined sub-classes are used.

Methods

addExclusiveAccessToAWTEvent(ResourceClient, UserEventRepository)

```
public boolean addExclusiveAccessToAWTEvent(org.davic.resources.ResourceClient client,
org.dvb.event.UserEventRepository userEvents)
```

An application should use this method to express its intend to have exclusive access to some events, but for these events to be received through the java.awt mechanism. The events the application wishes to receive are defined by the means of the UserEventRepository class. This repository is resolved at the time when this method call is made and adding or removing events from the repository after this method call does not affect the subscription to those events. An exclusive event will be sent to the application if this latest is focused.

The effect of multiple calls to this method by the same application with different instances of UserEventRepository shall be cumulative. If multiple calls to this method succeed in acquiring the events in the specified repositories then the semantics of each successful method call shall be obeyed as specified.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

NOTE: This method is intended for applications that wish to ensure particular input events are exclusively handled by a particular UI component - for example, ensuring that number keys go to a PIN number entry widget and nowhere else. Reservations made by this method are not automatically cancelled when the component or application loses focus. Hence if the reservation is not released, no other application will receive the reserved events - those events will be silently discarded. This is due to the absence of any requirement for the platform to provide end users with a visual indication of which application has focus. Without such an indication, if events were not silently discarded, the PIN code events could be received by the wrong application. Applications which have reserved input events using this method should monitor for loss of focus to a different application and change the appearence of the Component - e.g. to indicate that PIN entry is no longer happening. Once the appearance has changed, the application should release the exclusive access so that the previously reserved input events become available to other applications.

Parameters:

`client` - resource client.

`userEvents` - the user events the application wants to be inform of.

Returns:

true if the events defined in the repository have been acquired, false otherwise.

Throws:

`java.lang.IllegalArgumentException` - if the client argument is set to null.

addResourceStatusEventListener(ResourceStatusListener)

```
public void addResourceStatusEventListener(org.davic.resources.ResourceStatusListener listener)
```

Adds the specified resource status listener so that an application can be aware of any changes regarding exclusive access to some events.

Specified By:

`addResourceStatusEventListener` in interface `ResourceServer`

Parameters:

`listener` - the resource status listener.

addUserEventListener(UserEventListener, ResourceClient, UserEventRepository)

```
public boolean addUserEventListener(org.dvb.event.UserEventListener listener,
org.davic.resources.ResourceClient client, org.dvb.event.UserEventRepository userEvents)
```

Adds the specified listener to receive events coming from the user in an exclusive manner. The events the application wishes to receive are defined by the means of the UserEventRepository class. This repository is resolved at the time when this method call is made and adding or removing events from the repository after this method call does not affect the subscription to those events. The ResourceClient parameter indicates that the application wants to have an exclusive access to the user event defined in the repository.

The effect of multiple calls to this method by the same application with different instances of UserEventRepository shall be cumulative. If multiple calls to this method succeed in acquiring the events in the specified repositories then the semantics of each successful method call shall be obeyed as specified. Note that this can result in applications receiving the same event through more than one event listener.

Parameters:

`listener` - the listener to receive the user events.

`client` - resource client.

`userEvents` - a class which contains the user events it wants to be informed of.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

Returns:

true if the events defined in the repository have been acquired, false otherwise.

Throws:

`java.lang.IllegalArgumentException` - if the client argument is set to null.

addUserEventListener(UserEventListener, UserEventRepository)

```
public void addUserEventListener(org.dvb.event.UserEventListener listener,
org.dvb.event.UserEventRepository userEvents)
```

Adds the specified listener to receive events coming from the user. The events the application wishes to receive are defined by the means of the UserEventRepository class. This repository is resolved at the time when this method call is made and adding r removing events from the repository after this method call does not affect the subscription to those events.

The effect of multiple calls to this method by the same application with different instances of UserEventRepository shall be cumulative. If multiple calls to this method succeed in acquiring the events in the specified repositories then the semantics of each successful method call shall be obeyed as specified. Note that this can result in applications receiving the same event through more than one event listener.

Parameters:

`listener` - the listener to receive the user events.

`userEvents` - a class which contains the user events it wants to be informed of.

getInstance()

```
public static org.dvb.event.EventManager getInstance()
```

This method returns the sole instance of the EventManager class. The EventManager class is a singleton.

Returns:

the instance of the EventManager.

removeExclusiveAccessToAWTEvent(ResourceClient).

```
public void removeExclusiveAccessToAWTEvent(org.davic.resources.ResourceClient client)
```

The application should use this method to release its exclusive access to user events defined by the means of the addExclusiveAccessToAWTEvent method.

Parameters:

`client` - the client that is no longer interested in events previously registered.

removeResourceStatusEventListener(ResourceStatusListener)

```
public void removeResourceStatusEventListener(org.davic.resources.ResourceStatusListener listener)
```

Removes the specified resource status listener.

Specified By:

`removeResourceStatusEventListener` in interface `ResourceServer`

Parameters:

`listener` - the listener to remove.

removeUserEventListener(UserEventListener)

```
public void removeUserEventListener(org.dvb.event.UserEventListener listener)
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 398 of 861

Removes the specified listener so that it will no longer receives user events. If it is appropriate (i.e the application has asked for an exclusive access), the exclusive access is lost.

Parameters:

`listener` - the user event listener.

org.dvb.event

OverallRepository

Declaration

```
public class OverallRepository extends UserEventRepository
```

```
java.lang.Object
  |
  +--org.dvb.event.RepositoryDescriptor
        |
        +--org.dvb.event.UserEventRepository
              |
              +--org.dvb.event.OverallRepository
```

All Implemented Interfaces:

`org.davic.resources.ResourceProxy`

Description

This class defines a repository which initially contains all the user events which can be delivered to an application. This includes all keycodes for which KEY_PRESSED and KEY_RELEASED events can be generated and all keychars for which KEY_TYPED events can be generated. Note that the set of keycodes and keychars which can be generated is dependent on the input devices of the GEM terminal. For example, this pre-defined repository could be used by an application, which requires a pin code from the user, in order to prevent another applications from receiving events.

See Also:

`UserEvent`, `org.havi.ui.event.HKeyCapabilities`

Constructors

OverallRepository()

`public` OverallRepository()

The constructor for the repository. The name of the constructed instance (as returned by getName()) is implementation dependent.

OverallRepository(String)

```
public OverallRepository(java.lang.String name)
```

The constructor for the repository with a name.

Parameters:

`name` - the name to use for the repository

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238

org.dvb.event

RepositoryDescriptor

Declaration

```
public class RepositoryDescriptor implements org.davic.resources.ResourceProxy

java.lang.Object
  |
  +--org.dvb.event.RepositoryDescriptor
```

All Implemented Interfaces:

```
org.davic.resources.ResourceProxy
```

Direct Known Subclasses:

```
UserEventRepository
```

Description

An instance of this class will be sent to clients of the DVB event API to notify them (through the interface org.davic.resources.ResourceClient) when they are about to lose, or have lost, access to an event source. This object can be used by the application to get the name of the repository from which it will no longer be able to receive events. All instances of RepositoryDescriptor are also instances of UserEventRepository. This class is preserved for backwards compatibility with existing applications.

Methods

getClient()

```
public org.davic.resources.ResourceClient getClient()
```

Return the object which asked to be notified about withdrawal of the event source. This is the object passed as the `ResoourceClient` to whichever of the various 'add' methods on EventManager was used by the application to express interest in this repository.

Specified By:

`getClient` in interface `ResourceProxy`

Returns:

the object which asked to be notified about withdrawal of the event source. If the `UserEventRepository` has not yet been added to an `EventManager` then null shall be returned. Once the `UserEventRepository` has been added, the last used `ResourceClient` shall be returned even if the `UserEventRepository` has been since removed.

getName()

```
public java.lang.String getName()
```

Returns the name of the repository to which the lost, or about to be lost, user event belongs.

Returns:

String the name of the repository.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238