org.dvb.event

UserEvent

Declaration

```
public class UserEvent extends java.util.EventObject

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.event.UserEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Represents a user event. A user event is defined by a family, a type and either a code or a character. Unless stated otherwise, all constants used in the specification of this class are defined in `java.awt.event.KeyEvent` and its parent classes.

Fields

UEF_KEY_EVENT

```
public static final int UEF_KEY_EVENT
```

the family for events that are coming from the remote control or from the keyboard.

Constructors

UserEvent(Object, int, char, long)

```
public UserEvent(java.lang.Object source, int family, char keyChar, long when)
```

Constructor for a new UserEvent object representing a key being typed. This is the combination of a key being pressed and then being released. The type of UserEvents created with this constructor shall be KEY_TYPED. Key combinations which do not result in characters, such as keys like the red key on a remote control, shall not generate KEY_TYPED events. `KEY_TYPED` events shall have no modifiers and hence shall not report any modifiers as being down.

Parameters:

`source` - the `EventManager` which is the source of the event.

`family` - the event family.

`keyChar` - the character typed.

`when` - a long integer that specifies the time the event occurred.

Since:

MHP 1.0.1

UserEvent(Object, int, int, int, int, long)

```
public UserEvent(java.lang.Object source, int family, int type, int code, int modifiers, long when)
```

Constructor for a new UserEvent object representing a key being pressed.

Parameters:

`source` - the `EventManager` which is the source of the event.

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

`family` - the event family.

`type` - the event type. Either one of KEY_PRESSED or KEY_RELEASED.

`code` - the event code. One of the constants whose name begins in "VK_" defined in java.awt.event.KeyEvent or org.havi.ui.event.HRcEvent.

`modifiers` - the modifiers active when the key was pressed. These have the same semantics as modifiers in `java.awt.event.KeyEvent`.

`when` - a long integer that specifies the time the event occurred.

Methods

getCode()

```
public int getCode()
```

Returns the event code. For KEY_TYPED events, the code is VK_UNDEFINED.

Returns:

an int representing the event code.

getFamily()

```
public int getFamily()
```

Returns the event family. Could be UEF_KEY_EVENT.

Returns:

an int representing the event family.

getKeyChar()

```
public char getKeyChar()
```

Returns the character associated with the key in this event. If no valid Unicode character exists for this key event, keyChar is CHAR_UNDEFINED.

Returns:

a character.

Since:

MHP 1.0.1.

getModifiers()

```
public int getModifiers()
```

Returns the modifiers flag for this event. This method shall return 0 for UserEvents constructed using a constructor which does not include an input parameter specifying the modifiers.

Returns:

the modifiers flag for this event.

Since:

MHP 1.0.1.

getType()

```
public int getType()
```

Returns the event type. Could be KEY_PRESSED, KEY_RELEASED or KEY_TYPED.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 402 of 861

Returns:

an int representing the event type.

getWhen()

```
public long getWhen()
```

Returns the timestamp of when this event occurred.

Returns:

a long.

Since:

MHP 1.0.2.

isAltDown()

```
public boolean isAltDown()
```

Returns whether or not the Alt modifier is down on this event. This method shall return false for UserEvents constructed using a constructor which does not include an input parameter specifying the modifiers.

Returns:

whether the Alt modifier is down on this event.

Since:

MHP 1.0.1.

isControlDown()

```
public boolean isControlDown()
```

Returns whether or not the Control modifier is down on this event. This method shall return false for UserEvents constructed using a constructor which does not include an input parameter specifying the modifiers.

Returns:

whether the Control modifier is down on this event.

Since:

MHP 1.0.1.

isMetaDown()

```
public boolean isMetaDown()
```

Returns whether or not the Meta modifier is down on this event. This method shall return false for UserEvents constructed using a constructor which does not include an input parameter specifying the modifiers.

Returns:

whether the Meta modifier is down on this event.

Since:

MHP 1.0.1.

isShiftDown()

```
public boolean isShiftDown()
```

Returns whether or not the Shift modifier is down on this event. This method shall return false for UserEvents constructed using a constructor which does not include an input parameter specifying the modifiers.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 403 of 861

Returns:

whether the Shift modifier is down on this event.

Since:

MHP 1.0.1.

org.dvb.event

UserEventAvailableEvent

Declaration

```
public class UserEventAvailableEvent extends org.davic.resources.ResourceStatusEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.davic.resources.ResourceStatusEvent
              |
              +--org.dvb.event.UserEventAvailableEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

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. Where one change in user input event reservation results in instances of this event being sent to several applications, the following shall apply.

- Each application shall receive its own instance of the `UserEventRepository` object which forms the source to this event. Any changes made to that repository by any one application shall not impact the instance seen by any other application.
- Any application receiving an instance of this event is allowed to attempt to exclusively reserve some of the newly available user events. In this situation, the normal resource management policy of the platform as described elsewhere in the present document shall be obeyed.
- Any applications which have registered for shared access to any of these user events shall start receiving those events following receipt of this event.

Since:

MHP 1.0.2.

Constructors

UserEventAvailableEvent(Object)

```
public UserEventAvailableEvent(java.lang.Object source)
```

Constructor for the event.

Parameters:

`source` - a `UserEventRepository` which contains the events which stopped being exclusively reserved.

Since:

MHP 1.0.2.

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

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns a `UserEventRepository` which contains the events which were formerly exclusively reserved as passed into the constructor of the instance.

Overrides:

`getSource` in class `ResourceStatusEvent`

Returns:

a `UserEventRepository`.

Since:

MHP 1.0.2.

org.dvb.event
UserEventListener

Declaration

```
public interface UserEventListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

The listener interface for receiving user inputs.

Methods

userEventReceived(UserEvent)

```
public void userEventReceived(org.dvb.event.UserEvent e)
```

Called by the platform when a user input is received.

Parameters:

`e` - the user input event which was received.

org.dvb.event
UserEventRepository

Declaration

```
public class UserEventRepository extends RepositoryDescriptor
```

```
java.lang.Object
  |
  +--org.dvb.event.RepositoryDescriptor
        |
        +--org.dvb.event.UserEventRepository
```


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

All Implemented Interfaces:

`org.davic.resources.ResourceProxy`

Direct Known Subclasses:

`OverallRepository`

Description

The application will use this class to define the events that it wants to receive. Events that are able to be put in the repository are defined in the UserEvent class.

Where a repository includes a `KEY_PRESSED` type event without the `KEY_RELEASED` type event for the same key code or vice versa then exclusive reservations shall be made for both event types but only the one requested shall be received by the listener. Where a repository includes a `KEY_TYPED` event without the corresponding `KEY_PRESSED` and `KEY_RELEASED` events (excluding `KEY_PRESSED` or `KEY_RELEASED` events for modifiers), when an exclusive reservation is requested, it shall also be made for those corresponding `KEY_PRESSED` and `KEY_RELEASED` events but only the requested event shall be received by the listener.

Repositories do not keep a count of the number of times a particular user event is added or removed. Repeatedly adding an event to a repository has no effect. Removing an event removes it regardless of the number of times it has been added. For example, org.dvb.event.UserEventRepository.addUserEvent(UserEvent event) does nothing in case that the event is already in the repository. Events are considered to be already in the repository if an event with the same triplet of family, type and code is already in the repository.

If an application loses exclusive access to a repository, it shall lose access to all events defined in that repository. Repositories are resolved when they are passed into the methods of EventManager. Adding or removing events from the repository after those method calls does not affect the subscription to those events.

Unless stated otherwise, all constants used in the specification of this class are defined in `java.awt.event.KeyEvent` and its parent classes and not in this class.

See Also:

`UserEvent`

Constructors

UserEventRepository(String)

```
public UserEventRepository(java.lang.String name)
```

The method to construct a new UserEventRepository.

Parameters:

`name` - the name of the repository.

Methods

addAllArrowKeys()

```
public void addAllArrowKeys()
```

Adds the key codes for the arrow keys (VK_LEFT, VK_RIGHT, VK_UP, VK_DOWN). Any key codes already in the repository will not be added again. After calling this method, the keycodes shall be present for both the `KEY_PRESSED` and `KEY_RELEASED` modes.

addAllColourKeys()

```
public void addAllColourKeys()
```

Adds the key codes for the colour keys (VK_COLORED_KEY_0, VK_COLORED_KEY_1, VK_COLORED_KEY_2, VK_COLORED_KEY_3). Any key codes already in the repository will not be added again. After calling this method, the keycodes shall be present for both the `KEY_PRESSED` and `KEY_RELEASED` modes.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 406 of 861

addAllNumericKeys()

```
public void addAllNumericKeys()
```

Adds the key codes for the numeric keys (VK_0, VK_1, VK_2, VK_3, VK_4, VK_5, VK_6, VK_7, VK_8, VK_9). Any key codes already in the repository will not be added again. After calling this method, the keycodes shall be present for both the `KEY_PRESSED` and `KEY_RELEASED` modes.

addKey(int)

```
public void addKey(int keycode)
```

Adds the specified keycode to the repository. Keycodes added in this way shall be listed in the list of user events returned by the `getUserEvent` method. If a key is already in the repository, this method has no effect. After calling this method, the keycode shall be present for both the `KEY_PRESSED` and `KEY_RELEASED` modes.

Parameters:

`keycode` - the key code.

addUserEvent(UserEvent)

```
public void addUserEvent(org.dvb.event.UserEvent event)
```

Adds the given user event to the repository. The values of the modifiers (if any) in the `UserEvent` shall be ignored by the GEM terminal. The value of the source used to construct the specified `UserEvent` shall be ignored by the GEM terminal when the `UserEventRepository` is used to specify events which an application wants to receive.

Parameters:

`event` - the user event to be added in the repository.

getName()

```
public java.lang.String getName()
```

Returns the name of the current repository as passed to the constructor.

Overrides:

`getName` in class `RepositoryDescriptor`

Returns:

a String with the name of the repository.

getUserEvent()

```
public org.dvb.event.UserEvent[] getUserEvent()
```

Returns the list of the user events that are in the repository.

Returns:

an array which contains the user events that are in the repository.

removeAllArrowKeys()

```
public void removeAllArrowKeys()
```

Removes the key codes for the arrow keys (VK_LEFT, VK_RIGHT, VK_UP, VK_DOWN). Key codes from this set which are not present in the repository will be ignored. After calling this method, the keycodes shall not be present for both the `KEY_PRESSED` and `KEY_RELEASED` modes.

removeAllColourKeys()

```
public void removeAllColourKeys()
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 407 of 861

Removes the key codes for the colour keys (VK_COLORED_KEY_0, VK_COLORED_KEY_1, VK_COLORED_KEY_2, VK_COLORED_KEY_3). Key codes from this set which are not present in the repository will be ignored. After calling this method, the keycodes shall not be present for both the KEY_PRESSED and KEY_RELEASED modes.

removeAllNumericKeys()

```
public void removeAllNumericKeys()
```

Remove the key codes for the numeric keys (VK_0, VK_1, VK_2, VK_3, VK_4, VK_5, VK_6, VK_7, VK_8, VK_9). Key codes from this set which are not present in the repository will be ignored. After calling this method, the keycodes shall not be present for both the KEY_PRESSED and KEY_RELEASED modes.

removeKey(int)

```
public void removeKey(int keycode)
```

The method to remove a key from the repository. Removing a key which is not in the repository has no effect. After calling this method, the keycode shall not be present for both the KEY_PRESSED and KEY_RELEASED modes.

Parameters:

keycode - the key code.

removeUserEvent(UserEvent)

```
public void removeUserEvent(org.dvb.event.UserEvent event)
```

Remove a user event from the repository. Removing a user event which is not in the repository shall have no effect.

Parameters:

event - the event to be removed from the repository.

org.dvb.event

UserEventUnavailableEvent

Declaration

```
public class UserEventUnavailableEvent extends org.davic.resources.ResourceStatusEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.davic.resources.ResourceStatusEvent
              |
              +--org.dvb.event.UserEventUnavailableEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This event is sent to the resource status event listeners when user input events are exclusively reserved by an application.

Each application shall receive its own instance of the UserEventRepository object which forms the source to this event. Any changes made to that repository by any one application shall not impact the instance seen by any other application.

Any applications which have registered for shared access to any of these user events shall stop receiving those user events following receipt of this event. If such user events become available again, a UserEventAvailableEvent shall be generated by the platform before any more of those user events are received by applications.

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

Since:

MHP 1.0.2.

Constructors

UserEventUnavailableEvent(Object)

```
public UserEventUnavailableEvent(java.lang.Object source)
```

Constructor for the event.

Parameters:

`source` - a `UserEventRepository` which contains the events which were exclusively reserved.

Since:

MHP 1.0.2.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns a `UserEventRepository` which contains the events which were exclusively reserved as passed into the constructor of the instance.

Overrides:

`getSource` in class `ResourceStatusEvent`

Returns:

a `UserEventRepository`.

Since:

MHP 1.0.2.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 409 of 861

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

Package org.dvb.io.persistent

Description

Provides extensions to the java.io package for access to files held in persistent storage.

Class Summary	
Classes	
FileAccessPermissions	This class encapsulates file access permissions, world, Organisation and owner.
FileAttributes	This class encapsulates the attributes of a file stored in persistent storage.

org.dvb.io.persistent FileAccessPermissions

Declaration

```
public class FileAccessPermissions

java.lang.Object
  |
  +--org.dvb.io.persistent.FileAccessPermissions
```

Description

This class encapsulates file access permissions, world, Organisation and owner. World means all applications authorised to access persistent storage. Owner means the application which created the file. Organisation is defined as applications with the same organisation id as defined elsewhere in the present document.

Constructors

FileAccessPermissions(boolean, boolean, boolean, boolean, boolean, boolean)

```
public FileAccessPermissions(boolean readWorldAccessRight, boolean writeWorldAccessRight,
boolean readOrganisationAccessRight, boolean writeOrganisationAccessRight,
boolean readApplicationAccessRight, boolean writeApplicationAccessRight)
```

This constructor encodes all the file access permissions as a set of booleans.

Parameters:

`readWorldAccessRight` - read access for all applications.

`writeWorldAccessRight` - write access for all applications.

`readOrganisationAccessRight` - read access for organisation.

`writeOrganisationAccessRight` - write access for organisation.

`readApplicationAccessRight` - read access for the owner.

`writeApplicationAccessRight` - write access for the owner.

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

Methods

hasReadApplicationAccessRight()

```
public boolean hasReadApplicationAccessRight()
```

Query whether this permission includes read access for the owning application.

Returns:

true if the owning application can have read access, otherwise false.

hasReadOrganisationAccessRight()

```
public boolean hasReadOrganisationAccessRight()
```

Query whether this permission includes read access for the organisation

Returns:

true if applications in this organisation can have read access, otherwise false.

hasReadWorldAccessRight()

```
public boolean hasReadWorldAccessRight()
```

Query whether this permission includes read access for the world.

Returns:

true if all applications can have read access, otherwise false.

hasWriteApplicationAccessRight()

```
public boolean hasWriteApplicationAccessRight()
```

Query whether this permission includes write access for the owning application.

Returns:

true if the owning application can have write access, otherwise false.

hasWriteOrganisationAccessRight()

```
public boolean hasWriteOrganisationAccessRight()
```

Query whether this permission includes write access for the organisation.

Returns:

true if applications in this organisation can have read access, otherwise false.

hasWriteWorldAccessRight()

```
public boolean hasWriteWorldAccessRight()
```

Query whether this permission includes write access for the world.

Returns:

true if all applications can have write access, otherwise false.

setPermissions(boolean, boolean, boolean, boolean, boolean, boolean)

```
public void setPermissions(boolean ReadWorldAccessRight, boolean WriteWorldAccessRight,
boolean ReadOrganisationAccessRight, boolean WriteOrganisationAccessRight,
boolean ReadApplicationAccessRight, boolean WriteApplicationAccessRight)
```

This method allows to modify the permissions on this instance of the FileAccessPermission class.

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

Parameters:

`ReadWorldAccessRight` - read access for all applications.

`WriteWorldAccessRight` - write access for all applications.

`ReadOrganisationAccessRight` - read access for organisation.

`WriteOrganisationAccessRight` - write access for organisation.

`ReadApplicationAccessRight` - read access for the owner.

`WriteApplicationAccessRight` - write access for the owner.

org.dvb.io.persistent

FileAttributes

Declaration

```
public class FileAttributes

java.lang.Object
  |
  +--org.dvb.io.persistent.FileAttributes
```

Description

This class encapsulates the attributes of a file stored in persistent storage. The default attributes for a file are low priority, owner read/write only permissions and null expiration date.

Fields

PRIORITY_HIGH

```
public static final int PRIORITY_HIGH
```

Value for use as a file priority.

PRIORITY_LOW

```
public static final int PRIORITY_LOW
```

Value for use as a file priority.

PRIORITY_MEDIUM

```
public static final int PRIORITY_MEDIUM
```

Value for use as a file priority.

Constructors

FileAttributes(Date, FileAccessPermissions, int)

```
public FileAttributes(java.util.Date expiration_date, org.dvb.io.persistent.FileAccessPermissions p,
int priority)
```

Constructor.

Parameters:

`expiration_date` - an expiration date or null.

`p` - the access permissions to use.

`priority` - the priority to use in persistent storage.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 412 of 861

Methods

getExpirationDate()

```
public java.util.Date getExpirationDate()
```

Returns the expiration date. It will return the value used by the platform, which need not be the same as the value set.

Returns:

the expiration date.

getFileAttributes(File)

```
public static org.dvb.io.persistent.FileAttributes getFileAttributes(java.io.File f)
throws IOException
```

Get the attributes of a file.

Parameters:

`f` - the file to use.

Returns:

a copy of the attributes of a file.

Throws:

`java.lang.SecurityException` - if the application is denied access to the file or to directories needed to reach the file by security policy.

`java.io.IOException` - if access to the file fails due to an IO error or if the file reference is not to a valid location in persistent storage.

getPermissions()

```
public org.dvb.io.persistent.FileAccessPermissions getPermissions()
```

Returns the file access permissions.

Returns:

the file access permissions.

getPriority()

```
public int getPriority()
```

Returns the priority to use in persistent storage.

Returns:

the priority.

setExpirationDate(Date)

```
public void setExpirationDate(java.util.Date d)
```

Sets the expiration date. This field is a hint to the platform to identify the date after which a file is no longer useful as percieved by the application. The platform may choose to use a different date than the one given as a parameter.

Parameters:

`d` - the expiration date.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 413 of 861

setFileAttributes(FileAttributes, File)

```
public static void setFileAttributes(org.dvb.io.persistent.FileAttributes p, java.io.File f)
throws IOException
```

Associate a set of file attributes with a file.

Parameters:

`p` - the file attributes to use.

`f` - the file to use.

Throws:

`java.lang.SecurityException` - if the application is either denied access to the file or directories needed to reach the file by security policy or is not authorised to modify the attributes of the file.

`java.io.IOException` - if access to the file fails due to an IO error or if the file reference is not to a valid location in persistent storage

setPermissions(FileAccessPermissions)

```
public void setPermissions(org.dvb.io.persistent.FileAccessPermissions p)
```

Sets the file access permissions.

Parameters:

`p` - the file access permissions.

setPriority(int)

```
public void setPriority(int priority)
```

Sets the priority to use in persistent storage.

Parameters:

`priority` - the priority to set.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 414 of 861

Annex L (normative): User settings and preferences API

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

Package

org.dvb.user

Description

Provides access to settings and preferences configured by the end-user.

Class Summary	
Interfaces	
`UserPreferenceChangeListener`	An application wishing to be informed of any change to a user preference implements this interface.
Classes	
`Facility`	A facility maps a preference's name to a single value or to an array of values.
`GeneralPreference`	This class defines a set of general preferences.
`Preference`	This abstract class defines the Preference object.
`UserPreferenceChangeEvent`	This class defines the event sent to appropriate listeners when a user preference has been changed.
`UserPreferenceManager`	The UserPreferenceManager class gives access to the user preference settings.
`UserPreferencePermission`	This class is for user preference and setting permissions.
Exceptions	
`UnsupportedPreferenceException`	Thrown when a non-supported preference is used.

org.dvb.user

Facility

Declaration

```
public class Facility

java.lang.Object
  |
  +--org.dvb.user.Facility
```

Description

A facility maps a preference's name to a single value or to an array of values. A facility enables an application to define the list of values supported for a specified preference. For example, if an application is available in English or French then it can create a Facility ("User Language", {"English", "French"}). When the application will retrieve the "User Language" from the general preference it will specify the associated facility in order to get a Preference which will contain a set a values compatible with those supported by the application.

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

Constructors

Facility(String, String)

```
public Facility(java.lang.String preference, java.lang.String value)
```

Creates a Facility with a single value. This facility can be used by an application to retrieve a preference compatible with its capabilities.

Parameters:

`preference` - a String representing the name of the preference.

`value` - a String representing the value of the preference.

Facility(String, String[])

```
public Facility(java.lang.String preference, java.lang.String[] values)
```

Creates a Facility with a set of values. This facility can be used by an application to retrieve a preference compatible with its capabilities.

Parameters:

`preference` - a String representing the name of the preference.

`values` - an array of String representing the set of values.

org.dvb.user

GeneralPreference

Declaration

```
public final class GeneralPreference extends Preference
```

```
java.lang.Object
  |
  +--org.dvb.user.Preference
        |
        +--org.dvb.user.GeneralPreference
```

Description

This class defines a set of general preferences. These preferences are read from the receiver and each application (downloaded or not) can access them through the `UserPreferenceManager.read` method. The standardized preferences are "User Language", "Parental Rating", "User Name", "User Address", "User @", "Country Code", "Default Font Size", "Post Code".

When constructed, objects of this class are empty and have no values defined. Values may be added using the add methods inherited from the Preference class or by calling `UserPreferenceManager.read`.

The encodings of these standardized preferences are as follows:

- User Language: 3 letter ISO 639 [84] language codes.
- Parental Rating: string using the same encoding as returned by `javax.tv.service.guide.ContentRatingAdvisory.getDisplayText`.
- User Name: Name of the user. This shall be in an order that is appropriate for presentation directly to the user, e.g. in Western Europe, listing the first name first and the family name last is recommended as being culturally appropriate in many locales.
- User Address: postal address of the user, may contain multiple lines separated by carriage return characters (as defined in table 77).

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

- User @: e-mail address of the user in the SMTP form as defined in RFC 821 [71].
- Country Code: two letter ISO 3166-1 [25] country code.
- Default Font Size: preferred font size for normal body text expressed in points, decimal integer value encoded as a string (26 is the default; differing size indicates a preference of different font size than usual).
- "Post Code": no standard encoding is defined since the formats of post codes (US zip codes, German Postleitzahl) are normally country specific. The format used should be the most natural one for the country identified by the "Country Code" preference.

The preference names are treated as case-insensitive. The preference names shall be considered equal at least when the method java.lang.String.equalsIgnoreCase() returns true for the strings when the locale Locale.UK is used. Depending on the locale used in the implementation, implementations are allowed to consider equal also other upper and lower case character pairs in addition to those defined by the Locale.UK locale.

The standardized preference names in GEM shall only use such letters where the upper and lower case characters are recognized by the Locale.UK locale. Since the "Post Code" preference forms part of the "User Address" preference, successful calls to UserPreferenceManager.write(..) for a GeneralPreference("Post Code") shall modify the "User Address" preference. Hence UserPreferenceChangeEvents shall be generated for both "Post Code" and "User Address" in this specific situation. Successful calls to UserPreferenceManager.write(...) for a GeneralPreference("User Address") shall generate a UserPreferenceChangeEvent for "Post Code" if and only if the post code part of the address changes.

Constructors

GeneralPreference(String)

```
public GeneralPreference(java.lang.String name)
throws IllegalArgumentException
```

Constructs a GeneralPreference object. A general preference maps a preference name to a list of strings.

Parameters:

`name` - the general preference name.

Throws:

`java.lang.IllegalArgumentException` - if the preference's name is not supported.

org.dvb.user

Preference

Declaration

```
public abstract class Preference

java.lang.Object
   |
   +--org.dvb.user.Preference
```

Direct Known Subclasses:

`GeneralPreference`

Description

This abstract class defines the Preference object. A Preference maps a name to a list of favourite values. The first element in the list is the favourite value for this preference.

The preference names are treated as case-insensitive. The preference names shall be considered equal at least when the method java.lang.String.equalsIgnoreCase() returns true for the strings when the locale java.util.Locale.UK is used. Depending on the locale used in the implementation, implementations are allowed to consider equal also other upper and lower case character pairs in addition to those defined by the Locale.UK locale.

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

The standardized preference names in GEM shall only use such letters where the upper and lower case characters are recognized by the Locale.UK locale.

Constructors

Preference()

```
protected Preference()
```

This protected constructor is only present to enable sub-classes of this one to be defined by the platform. It is not intended to be used by inter-operable applications.

Preference(String, String)

```
public Preference(java.lang.String name, java.lang.String value)
```

Creates a new preference with the specified name and the specified value. This single value will be the favourite one for this preference.

Parameters:

`name` - a String object representing the name of the preference.

`value` - a String object representing the value of the preference.

Preference(String, String[])

```
public Preference(java.lang.String name, java.lang.String[] value)
```

Creates a new preference with the specified name and the specified value set. Each value in the value set must appear only once. The behaviour if a value is duplicated is implementation dependent.

Parameters:

`name` - a String object representing the name of the preference.

`value` - an array of String objects representing the set of values for this preference ordered from the most favourite to the least favourite.

Methods

add(int, String)

```
public void add(int position, java.lang.String value)
```

Adds a new value for this preference. The value is inserted at the specified position. If the value is already in the list then it is moved to the position specified. 1 If the position is greater than or equal to the length of the list, then the value is added to the end of this list. If the position is negative, then the value is added to the beginning of this list.

Parameters:

`position` - an int representing the position in the list counting from zero.

`value` - a String representing the new value to insert.

add(String)

```
public void add(java.lang.String value)
```

Adds a new value for this preference. The value is added to the end of the list. If the value is already in the list then it is moved to the end of the list.

Parameters:

`value` - a String object representing the new value.

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

add(String[])

```
public void add(java.lang.String[] values)
```

Adds several new values for this preferences. The values are added to the end of the list in the same order as they are found in the array passed to this method. Any values already in the list are moved to the position in the list which they would have if they were not already present.

Parameters:

`values` - an array of strings representing the values to add.

Since:

MHP 1.0.1.

getFavourites()

```
public java.lang.String[] getFavourites()
```

Returns the list of favourite values for this preference. Returns an empty array if no value sets are defined for this preference.

Returns:

an array of String representing the favourite values for this preference.

getMostFavourite()

```
public java.lang.String getMostFavourite()
```

Returns the most favourite value for this preference, that is, the first element of the list.

Returns:

a String representing the most favourite value for this preference. Returns null if no value is defined for this preference.

getName()

```
public java.lang.String getName()
```

Returns the name of the preference.

Returns:

a String object representing the name of the preference.

getPosition(String)

```
public int getPosition(java.lang.String value)
```

Returns the position in the list of the specified value.

Parameters:

`value` - a String representing the value to look for.

Returns:

an integer representing the position of the value in the list counting from zero. If the value is not found then it returns -1.

hasValue()

```
public boolean hasValue()
```

Tests if this preference has at least one value set.

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

Returns:

true if this preference has at least one value set, false otherwise.

remove(String)

```
public void remove(java.lang.String value)
```

Removes the specified value from the list of favourites. If the value is not in the list then the method call has no effect.

Parameters:

`value` - a String representing the value to remove.

removeAll()

```
public void removeAll()
```

Removes all the values of a preference

Since:

MHP 1.0.1.

setMostFavourite(String)

```
public void setMostFavourite(java.lang.String value)
```

Sets the most favourite value for this preference. If the value is already in the list, then it is moved to the head. If the value is not already in the list then it is added at the head.

Parameters:

`value` - the most favourite value.

toString()

```
public java.lang.String toString()
```

Convert name and favourites to a String.

Overrides:

`toString` in class `Object`.

Returns:

the preference name and favourites.

org.dvb.user

UnsupportedPreferenceException

Declaration

```
public class UnsupportedPreferenceException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.user.UnsupportedPreferenceException
```

All Implemented Interfaces:

```
java.io.Serializable
```


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

Description

Thrown when a non-supported preference is used.

Constructors

UnsupportedPreferenceException()

```
public UnsupportedPreferenceException()
```

Constructs a `UnsupportedPreferenceException` with no detail message.

UnsupportedPreferenceException(String)

```
public UnsupportedPreferenceException(java.lang.String a)
```

Constructs a `UnsupportedPreferenceException` with a detail message.

Parameters:

`a` - the detail message.

org.dvb.user

UserPreferenceChangeEvent

Declaration

```
public class UserPreferenceChangeEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.user.UserPreferenceChangeEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This class defines the event sent to appropriate listeners when a user preference has been changed.

Constructors

UserPreferenceChangeEvent(String)

```
public UserPreferenceChangeEvent(java.lang.String preferenceName)
```

Constructs a new event.

Parameters:

`preferenceName` - the name of the modified preference.

Methods

getName()

```
public java.lang.String getName()
```

Returns the name of the modified Preference.

Returns:

the Preference name.

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

org.dvb.user

UserPreferenceChangeListener

Declaration

```
public interface UserPreferenceChangeListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

An application wishing to be informed of any change to a user preference implements this interface.

Methods

receiveUserPreferenceChangeEvent(UserPreferenceChangeEvent)

```
public void receiveUserPreferenceChangeEvent(org.dvb.user.UserPreferenceChangeEvent e)
```

This method is called when a user preference changes.

Parameters:

`e` - the event notifying this event.

org.dvb.user

UserPreferenceManager

Declaration

```
public class UserPreferenceManager
```

```
java.lang.Object
  |
  +--org.dvb.user.UserPreferenceManager
```

Description

The UserPreferenceManager class gives access to the user preference settings. This class provides a set of methods that allow an application to read or save user settings. It also provides a mechanism to notify applications when a preference has been modified. The value of a user setting, retrieved with the read method, is a copy of the value that is stored in the receiver. The write method, if authorized, overwrites the stored value.

NOTE: GEM implementations are not required to validate the values in Preference objects, even those which are saved using the write method. Applications with write permissions need to be very careful that the values written are valid. Applications reading permissions need to be aware of the possibility that a previous application has set an invalid value.

Methods

addUserPreferenceChangeListener(UserPreferenceChangeListener)

```
public void addUserPreferenceChangeListener(org.dvb.user.UserPreferenceChangeListener l)
```

Adds a listener for changes in user preferences as held in the MHP terminal. Specifically this includes changes made by MHP applications succeeding in calling the write() method on this class. If the implementation of the MHP terminal allows the end user to change preferences then these changes also includes changes made to preferences by this mechanism. It does not include changes made to a Preference instance within the scope of a single MHP application.

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

Parameters:

`l` - the listener to add.

getInstance()

```
public static org.dvb.user.UserPreferenceManager getInstance()
```

Return an instance of the `UserPreferenceManager` for this application. Repeated calls to this method by the same application shall return the same instance.

Returns:

an instance of `UserPreferenceManager`.

read(Preference)

```
public void read(org.dvb.user.Preference p)
```

Allows an application to read a specified user preference. When end-user preferences are read into a `Preference` object from the MHP terminal, the ordering of these values shall be as determined by the end-user, from most preferred to least preferred to the extent that this is known.

Parameters:

`p` - an object representing the preference to read.

Throws:

`java.lang.SecurityException` - if the calling application is denied access to this preference.

read(Preference, Facility)

```
public void read(org.dvb.user.Preference p, org.dvb.user.Facility facility)
```

Allows an application to read a specified user preference taking into account the facility defined by the application. After this method returns, the values in the `Preference` object shall be the values of that user preference with any unsupported values from the `Facility` removed from that list. Note that the order of values returned here need not be the same as that returned by `read(Preference)`.

If the intersection between the two sets of values is empty then the preference will have no value. If there is a mis-match between the name of the preference used when constructing the facility and the name of the preference used in this method then the preference will have no value.

Parameters:

`p` - an object representing the preference to read.

`facility` - the preferred values of the application for the preference

Throws:

`java.lang.SecurityException` - if the calling application is denied access to this preference.

removeUserPreferenceChangeListener(UserPreferenceChangeListener)

```
public void removeUserPreferenceChangeListener(org.dvb.user.UserPreferenceChangeListener l)
```

Removes a listener for changes in user preferences.

Parameters:

`l` - the listener to remove.

write(Preference)

```
public void write(org.dvb.user.Preference p)
throws UnsupportedPreferenceException, IOException
```


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

Saves the specified user preference. If this method succeeds then it will change the value of this preference for all future MHP applications.

Parameters:

`p` - the preference to save.

Throws:

`UnsupportedPreferenceException` - if the preference provided is not a standardized preference as defined for use with `GeneralPreference`.

`java.lang.SecurityException` - if the application does not have permission to call this method.

`java.io.IOException` - if saving the preference fails for other I/O reasons.

org.dvb.user

UserPreferencePermission

Declaration

```
public class UserPreferencePermission extends java.security.BasicPermission
```

```
java.lang.Object
  |
  +--java.security.Permission
        |
        +--java.security.BasicPermission
              |
              +--org.dvb.user.UserPreferencePermission
```

All Implemented Interfaces:

`java.security.Guard`, `java.io.Serializable`

Description

This class is for user preference and setting permissions. A UserPreferencePermission contains a name, but no actions list.

The permission name can either be "read" or "write". The "read" permission allows an application to read the user preferences and settings (using `UserPreferenceManager.read`) for which read access is not always granted. Access to the following settings/preferences is always granted: "User Language", "Parental Rating", "Default Font Size" and "Country Code".

The "write" permission allows an application to modify user preferences and settings (using `UserPreference-Manager.write`).

Constructors

UserPreferencePermission(String)

```
public UserPreferencePermission(java.lang.String name)
```

Creates a new UserPreferencePermission with the specified name. The name is the symbolic name of the User-PreferencePermission.

Parameters:

`name` - the name of the UserPreferencePermission.

UserPreferencePermission(String, String)

```
public UserPreferencePermission(java.lang.String name, java.lang.String actions)
```


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

Creates a new UserPreferencePermission object with the specified name. The name is the symbolic name of the User-PreferencePermission, and the actions String is unused and should be null. This constructor exists for use by the Policy object to instantiate new Permission objects.

Parameters:

`name` - the name of the UserPreferencePermission.

`actions` - should be null.

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

Annex M (informative): SI access API

GEM does not specify a particular SI API (such as DVB SI), but nonetheless, a functionally equivalent "SI" is required for some profiles.

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

Annex N (normative): Streamed media API extensions

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

N.1 Active Format Definition

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

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

N.1.1 MHP Signalling for Active Format Definition

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

N.1.2 Drip-feed APIs

For GEM terminal specifications, which don't require MPEG-2 Video "drips", the corresponding APIs in `org.dvb.media.DripFeedPermission` and `org.dvb.media.DripFeedDataSource` are not required.

N.1.3 VideoFormatControl

Signalling for the active format description as defined in annex B of TS 101 154 [2] is optional in GEM.

Package org.dvb.media

Description

Provides DVB specific extensions to the Java Media Framework.

Class Summary	
Interfaces	
`BackgroundVideoPresentationControl`	A control to support the setting and querying of the video presentation for background players.
`ComponentBasedPlayerControl`	The presence of this Control on a Player indicates that the Player can be used as either a background Player or a component-based Player.
`DVBMediaSelectControl`	DVBMediaSelectControl extends `MediaSelectControl` allowing the selection of different kinds of content in a running `Player`.
`SubtitleListener`	Report that a subtitle event has happened.

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

Class Summary	
SubtitlingEventControl	Allow applications to register and unregister their interest in events related to the availability and presentation of subtitles.
VideoFormatControl	This provides a means for applications to get information associated with the format and aspect ratio of the video being presented to the user.
VideoFormatListener	The listener used to receive video format events
VideoPresentationControl	A control to support setting and querying the video presentation.
Classes	
ActiveFormatDescriptionChangedEvent	Event signalling that the transmitted active format definition has changed
AspectRatioChangedEvent	Event signalling that the aspect ratio of the transmitted video has changed
CAStopEvent	This event is generated whenever access to a service is withdrawn by the CA system, e.g. at the end of a free preview period.
Codecs	Class providing access to information on video and audio formats that a GEM terminal can decode.
DFCChangedEvent	Event signalling that the decoder format conversion being used has changed
DripFeedDataSource	This class allows to create a source for a JMF player to be able to feed the decoder progressively with parts of a clip (e.g.
DripFeedPermission	This class represents a permission to access the drip feed mode.
NoComponentSelectedEvent	This event is generated whenever presentation of a stream stops because there are no selected components to present.
PresentationChangedEvent	This event is generated whenever the content being presented by a player changes for reasons outside the control of the application.
ServiceRemovedEvent	This event is generated whenever access to a service stops because the service concerned has been removed from the network.
StopByResourceLossEvent	This event is generated whenever presentation of a stream stops because the player has lost so many resources that it cannot continue.
SubtitleAvailableEvent	Report that subtitles are available to be presented having been unavailable.
SubtitleNotAvailableEvent	Inform an application that a subtitle stream has vanished from the network.
SubtitleNotSelectedEvent	Report that subtitles are not now selected.
SubtitleSelectedEvent	Report that subtitles are now selected.
VideoFormatEvent	The base class for all other events relating to changes in video format
VideoTransformation	VideoTransformation objects express video transformations, i.e.
Exceptions	
CAException	This exception is thrown when access to a media stream is denied by the CA system.

org.dvb.media

ActiveFormatDescriptionChangedEvent

Declaration

```
public class ActiveFormatDescriptionChangedEvent extends VideoFormatEvent

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.media.VideoFormatEvent
              |
              +--org.dvb.media.ActiveFormatDescriptionChangedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Event signalling that the transmitted active format definition has changed.

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

Constructors

ActiveFormatDescriptionChangedEvent(Object, int)

```
public ActiveFormatDescriptionChangedEvent(java.lang.Object source, int newFormat)
```

Construct the event.

Parameters:

`source` - the source of the event.

`newFormat` - the new active format description.

Methods

getNewFormat()

```
public int getNewFormat()
```

Get the new active format description.

Returns:

the new active format description. The value of this is represented by one of the constants from `VideoFormatControl` and shall be the value passed into the constructor of the event.

org.dvb.media AspectRatioChangedEvent

Declaration

```
public class AspectRatioChangedEvent extends VideoFormatEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.media.VideoFormatEvent
              |
              +--org.dvb.media.AspectRatioChangedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Event signalling that the aspect ratio of the transmitted video has changed.

Constructors

AspectRatioChangedEvent(Object, int)

```
public AspectRatioChangedEvent(java.lang.Object source, int newRatio)
```

Construct the event.

Parameters:

`source` - the source of the event.

`newRatio` - the new aspect ratio of the transmitted video.

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

Methods

getNewRatio()

```
public int getNewRatio()
```

Get the new aspect ratio of the transmitted video.

Returns:

the new aspect ratio of the video. The value of this is represented by one of the constants from the VideoFormatControl class and shall be the value passed into the constructor of the event.

org.dvb.media BackgroundVideoPresentationControl

Declaration

```
public interface BackgroundVideoPresentationControl extends VideoPresentationControl
```

All Superinterfaces:

```
javax.media.Control, VideoPresentationControl
```

Description

A control to support the setting and querying of the video presentation for background players.

Methods

getClosestMatch(VideoTransformation)

```
public org.dvb.media.VideoTransformation getClosestMatch(org.dvb.media.VideoTransformation t)
```

This method takes a video transformation and returns the closest match of that video transformation that can be supported for the currently selected video. If the input video transformation can be supported, then the output video transformation will have the same parameters as the input video transformation. The definition of 'closest match' is implementation dependent.

Parameters:

`t` - the input video transformation.

Returns:

the closest match to the input video transformation. If the input video transformation is supported, then the input video transformation will be returned (the same instance), otherwise a newly created instance will be returned.

getVideoTransformation()

```
public org.dvb.media.VideoTransformation getVideoTransformation()
```

Return the current video transformation.

Returns:

the video transformation (clipping/scaling/positioning) that is currently used for displaying the video.

setVideoTransformation(VideoTransformation)

```
public boolean setVideoTransformation(org.dvb.media.VideoTransformation t)
```

Sets a new video transformation (clipping/scaling/positioning). If the new video transformation is not supported, then the video transformation will not be changed at all (no best effort attempt is made).

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

Parameters:

`t` - the new video transformation.

Returns:

true if the video transformation is supported and has been set, false otherwise.

org.dvb.media

CAException

Declaration

```
public class CAException extends java.io.IOException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.media.CAException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This exception is thrown when access to a media stream is denied by the CA system. It will typically be thrown by calls to DataSource.start() when access to the stream accessed by the DataSource is denied.

Constructors

CAException()

```
public CAException()
```

Constructor without a reason.

CAException(String)

```
public CAException(java.lang.String reason)
```

Constructor with a reason.

Parameters:

`reason` - the reason why access to the stream failed.

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

org.dvb.media

CAStopEvent

Declaration

```
public class CAStopEvent extends javax.media.StopEvent

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--javax.media.ControllerEvent
              |
              +--javax.media.TransitionEvent
                    |
                    +--javax.media.StopEvent
                          |
                          +--org.dvb.media.CAStopEvent
```

All Implemented Interfaces:

`javax.media.MediaEvent`, `java.io.Serializable`

Description

This event is generated whenever access to a service is withdrawn by the CA system, e.g. at the end of a free preview period. It is not generated when an attempt to construct a Player or DataSource fails due to CA restrictions, or when only some of the presented content is not available or alternate content is presented. Generation of this event informs the application that the Player is no longer presenting any content.

Constructors

CAStopEvent(Controller)

```
public CAStopEvent(javax.media.Controller source)
```

Construct an event.

Parameters:

`source` - the controller which was presenting the service.

CAStopEvent(Controller, int, int, int, MediaLocator)

```
public CAStopEvent(javax.media.Controller source, int previous, int current, int target,
javax.media.MediaLocator stream)
```

Construct an event.

Parameters:

`source` - the controller which was presenting the service.

`stream` - the locator of the stream from which access has been withdrawn.

`previous` - the previous state of the controller.

`current` - the current state of the controller.

`target` - the target state of the controller.

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

Methods

getStream()

```
public javax.media.MediaLocator getStream()
```

This method returns the stream from which access has been withdrawn.

Returns:

the locator for the stream concerned.

org.dvb.media

Codecs

Declaration

```
public class Codecs
```

```
java.lang.Object
  |
  +--org.dvb.media.Codecs
```

Description

Class providing access to information on video and audio formats that a GEM terminal can decode.

Since:

MHP 1.2.

Methods

getAudioFormats()

```
public static java.lang.String[] getAudioFormats()
```

Return the list of audio formats that can be decoded. Formats are encoded as MPEG-7 term IDs as extended by the DVB IPI handbook. e.g. MPEG-1 layer 3 audio is encoded as "3.3".

Returns:

an array of audio formats.

getVideoFormats()

```
public static java.lang.String[] getVideoFormats()
```

Return the list of video formats that can be decoded. Formats are encoded as MPEG-7 term IDs as extended by the DVB IPI handbook. e.g. MPEG-2 video, main profile, main level is encoded as "2.2.2".

Returns:

an array of video formats.

isAudioFormatSupported(String)

```
public static boolean isAudioFormatSupported(java.lang.String term)
```

Query if a specific audio format can be decoded. Formats are encoded as MPEG-7 term IDs as extended by the DVB IPI handbook. e.g. MPEG-1 layer 3 audio is encoded as "3.3".

Parameters:

`term` - a audio format.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 433 of 861

Returns:

true if the specified format can be decoded otherwise false.

isVideoFormatSupported(String)

```
public static boolean isVideoFormatSupported(java.lang.String term)
```

Query if a specific video format can be decoded. Formats are encoded as MPEG-7 term IDs as extended by the DVB IPI handbook. e.g. MPEG-2 video, main profile, main level is encoded as "2.2.2".

Parameters:

`term` - a video format.

Returns:

true if the specified format can be decoded otherwise false.

org.dvb.media

ComponentBasedPlayerControl

Declaration

```
public interface ComponentBasedPlayerControl extends javax.media.Control
```

All Superinterfaces:

```
javax.media.Control
```

Description

The presence of this Control on a Player indicates that the Player can be used as either a background Player or a component-based Player. Players for broadcast streaming formats that do not provide this control will always return null from their getVisualComponent() method.

A Player that provides this control must continue to provide an instance of it (not necessarily the same instance) after any transitions from background to component-based modes and vice-versa.

Methods

releaseVisualComponent()

```
public boolean releaseVisualComponent()
throws IllegalStateException
```

Transition the associated Player from a component-based Player to a background Player. Also releases the ownership of the visual component to allow other applications to acquire it.

The calling application must own the visual component - i.e. it must have obtained a non-null visual component through Player.getVisualComponent() on the Player associated with this control.

If the application has called paint() on that visual component, then this method transitions the Player from a component-based Player to a background Player, and releases the visual component. Otherwise, the Player is still a background Player so this method merely releases the visual component.

If this call transitions the Player from a component-based Player to a background Player, then the video is intially sized and positioned as close to the last location of the visual component as possible, given the limitations exposed after the transition by org.dvb.media.BackgroundVideoPresentationControl .getClosestMatch() or (equivalently) javax.tv.media.AWTVideoSizeControl.checkSize().

After the 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.

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

The application should ensure that the visual component is removed from the AWT hierarchy before making this call. After this call, interoperable applications should not make any calls to the visual component aquired before this call. However, such calls will not affect the Player in any way. (For example, calling paint() on that visual component will not cause the Player to become a component-based player). The appearence of the visual component if it is displayed after this call is implementation-dependant.

Future calls to Player.getVisualComponent() will return a visual component that is different from the visual component returned before this call.

Returns:

true iff the Player has changed from component-based to background mode.

Throws:

`java.lang.IllegalStateException` - If the calling application does not own the visual component.

swapVideoComponent(Player)

```
public void swapVideoComponent(javax.media.Player other)
```

Swap the presentation of the video for this JMF player and another Player between a component and the background.

Parameters:

`other` - the other player to use in the swap.

Throws:

`java.lang.IllegalArgumentException` - if both this player and the other player are background players or if both this player and the other player are component players or if the calling application does not have a reference to the java.awt.Component used in the swap or if either player is not presenting video

Since:

MHP 1.1.3.

org.dvb.media

DFCChangedEvent

Declaration

```
public class DFCChangedEvent extends VideoFormatEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.media.VideoFormatEvent
              |
              +--org.dvb.media.DFCChangedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Event signalling that the decoder format conversion being used has changed.

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

Constructors

DFCChangedEvent(Object, int)

```
public DFCChangedEvent(java.lang.Object source, int newDFC)
```

Construct the event.

Parameters:

`source` - the source of the event.

`newDFC` - the new decoder format conversion being used.

Methods

getNewDFC()

```
public int getNewDFC()
```

Get the new decoder format conversion.

Returns:

the new decoder format conversion. The value of this is represented by one of the constants from the VideoFormatControl class and shall be the value passed into the constructor of the event.

org.dvb.media

DripFeedDataSource

Declaration

```
public class DripFeedDataSource extends javax.media.protocol.DataSource
```

```
java.lang.Object
  |
  +--javax.media.protocol.DataSource
        |
        +--org.dvb.media.DripFeedDataSource
```

All Implemented Interfaces:

```
javax.media.protocol.Controls, javax.media.Duration
```

Description

This class allows to create a source for a JMF player to be able to feed the decoder progressively with parts of a clip (e.g. I or P MPEG-2 frame) according to the drip-fed mode format defined in the GEM content format clause.

To start using the drip-feed mode, the application needs to instantiate a player representing a MPEG-2 video decoder and have its source be a DripFeedDataSource instance.

A DripFeedDataSource instance can be obtained by calling the default constructor of the class.

A player that will be bound to a MPEG-2 video decoder (when realized) can be created with a locator of the following text representation: "dripfeed://".

After having the DripFeedDataSource connected to a Player representing a MPEG-2 video decoder, the following rules applies:

- If the feed method is called when the player is in the "prefetched" state the image will be stored so that when the player goes in the "started" state it will be automatically displayed.
- If the feed method is called when the player is in the "started" mode, the frame shall be displayed immediately. In particular it is not required to feed a second frame to the decoder to display the first frame.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 436 of 861

- If the feed method is called when the player is in any other state (or if the DripFeedDataSource is not connected to a player), it will be ignored by the platform implementation.

Constructors

DripFeedDataSource()

```
public DripFeedDataSource()
```

Constructor. A call to the constructor will throw a security exception if the application is not granted the right to use the drip feed mode. The constructor shall automatically set the MediaLocator for this DataSource to the only allowed value: dripfeed://. There is no need for applications to later call setLocator.

Methods

connect()

```
public void connect()
throws IOException
```

This method shall not be used and has no effect. This source is considered as always connected.

Overrides:

`connect` in class `DataSource`.

Throws:

`java.io.IOException` - never thrown in this sub-class.

disconnect()

```
public void disconnect()
```

This method shall not be used and has no effect. This source is considered as always connected.

Overrides:

`disconnect` in class `DataSource`

feed(byte[])

```
public void feed(byte[] clip_part)
```

This method allows an application to feed the decoder progressively with parts of a clip (e.g. I or P MPEG-2 frame) according to the drip-fed mode format defined in the GEM content format clause.
The feed method shall not be called more often than every 500ms. If this rule is not respected, display is not guaranteed.

While in the prefetch state the drip feed data source is only required to correctly process a single invocation of this method where the data consists only of a single I frame. Possible additional invocations while in the prefetch state shall have implementation specific results.

Parameters:

`clip_part` - Chunk of bytes compliant with the drip-fed mode format defined in the GEM content format clause (i.e. one MPEG-2 frame with optional synctactic MPEG-2 elements).

getContentType()

```
public java.lang.String getContentType()
```

This method shall return the content type for mpeg-2 video "drips".

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

Overrides:

`getContentType` in class `DataSource`.

Returns:

the content type for MPEG-2 video drips.

getControl(String)

```
public java.lang.Object getControl(java.lang.String controlType)
```

Obtain the object that implements the specified Class or Interface. The full class or interface name must be used. If the control is not supported then null is returned.

Overrides:

`getControl` in class `DataSource`.

Parameters:

`controlType` - the full class or interface name of the requested control.

Returns:

the object that implements the control, or null.

getControls()

```
public java.lang.Object[] getControls()
```

Obtain the collection of objects that control this object. If no controls are supported, a zero length array is returned.

Overrides:

`getControls` in class `DataSource`.

Returns:

the collection of object controls.

getDuration()

```
public javax.media.Time getDuration()
```

This method shall not be used and has no effect.

Overrides:

`getDuration` in class `DataSource`

Returns:

DURATION_UNKNOWN.

start()

```
public void start()
throws IOException
```

This method shall not be used and has no effect. This source is considered as always started.

Overrides:

`start` in class `DataSource`

Throws:

`java.io.IOException` - never thrown in this sub-class.

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

stop()

```
public void stop()
throws IOException
```

This method shall not be used and has no effect. This source is considered as always started.

Overrides:

`stop` in class `DataSource`

Throws:

`java.io.IOException` - never thrown in this sub-class.

org.dvb.media

DripFeedPermission

Declaration

```
public class DripFeedPermission extends java.security.BasicPermission
```

```
java.lang.Object
  |
  +--java.security.Permission
        |
        +--java.security.BasicPermission
              |
              +--org.dvb.media.DripFeedPermission
```

All Implemented Interfaces:

`java.security.Guard`, `java.io.Serializable`

Description

This class represents a permission to access the drip feed mode.

Constructors

DripFeedPermission(String)

```
public DripFeedPermission(java.lang.String name)
```

Create a new DripFeedPermission.

Parameters:

`name` - the name string is currently unused and should be empty.

DripFeedPermission(String, String)

```
public DripFeedPermission(java.lang.String name, java.lang.String actions)
```

Create a new DripFeedPermission. This constructor is used by the policy class to instantiate new permission objects.

Parameters:

`name` - The name string is currently unused and should be empty.

`actions` - The actions string is currently unused and should be null.

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

Methods

implies(Permission)

```
public boolean implies(java.security.Permission p)
```

Checks if the specified permission is "implied" by this object.

Since name and actions are not used, the only check needed is whether p is also a DripFeedPermission.

Overrides:

`implies` in class `BasicPermission`.

Parameters:

`p` - the permission to check against.

Returns:

true if the passed permission is equal to or implied by this permission, false otherwise.

org.dvb.media

DVBMediaSelectControl

Declaration

```
public interface DVBMediaSelectControl extends javax.tv.media.MediaSelectControl
```

All Superinterfaces:

`javax.media.Control`, `javax.tv.media.MediaSelectControl`

Description

DVBMediaSelectControl extends `MediaSelectControl` allowing the selection of different kinds of content in a running `Player`. The extension is to allow the selection in a single operation of all the media service components in a service without needing knowledge about which media service components are present in that service.

Since:

MHP 1.0.2.

See Also:

`javax.tv.media.MediaSelectControl`

Methods

add(Locator, StreamType)

```
public void add(javax.tv.locator.Locator component, javax.tv.service.navigation.StreamType type)
throws InvalidLocatorException, InvalidServiceComponentException, InsufficientResourcesException, Se
curityException
```

Adds a service component (for example, subtitles) to the presentation over-riding the stream type signalled for the component. This is an asynchronous operation that is completed on receipt of a MediaSelectEvent. Components whose addition would require Player resynchronization are not permitted. If the specified service component is already part of the presentation, this method does nothing.

Parameters:

`component` - The locator representing an individual service component to add to the presentation.

`type` - The stream type of the component.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 440 of 861

Throws:

`javax.tv.locator.InvalidLocatorException` - If the specified locator does not reference a selectable service component.

`javax.tv.service.selection.InvalidServiceComponentException` - If the addition of the service component would require resynchronization of the Player, if the service component is not part of the Service to which the MediaSelectControl is restricted, or if the service component must be presented in conjunction with another service component that is not part of the current presentation.

`javax.tv.service.selection.InsufficientResourcesException` - - If the operation cannot be completed due to a lack of system resources.

`java.lang.SecurityException` - - If the caller does not have MediaSelectPermission(component) permission.

Since:

MHP 1.1.2.

select(Locator[], StreamType[])

```
public void select(javax.tv.locator.Locator[] components,
javax.tv.service.navigation.StreamType[] types)
throws InvalidLocatorException, InvalidServiceComponentException, InsufficientResourcesException, Se
curityException
```

Selects one or more service components for presentation over-riding the stream type signalled for the component. If some content is currently playing, it is replaced in its entirety by the specified selection. This is an asynchronous operation that is completed on receipt of a MediaSelectEvent. Note that for certain selections that imply a different time base or otherwise change synchronization relationships, a RestartingEvent will be posted by the Player. If any of the components are successfully presented then a MediaSelectSucceededEvent is generated with the locator array containing only those components that were successfully presented. A MediaSelectFailedEvent is only generated if none of the components are successfully presented.

Parameters:

`components` - An array of locators representing a set of individual service components to present together.

`types` - The stream type corresponding to each locator.

Throws:

`javax.tv.locator.InvalidLocatorException` - If a locator provided does not reference a selectable service component.

`javax.tv.service.selection.InvalidServiceComponentException` - If a specified service component is not part of the Service to which the MediaSelectControl is restricted, if a specified service component must be presented in conjunction with another service component not contained in components, if the specified set of service components cannot be presented as a coherent whole, or if the service components are not all available simultaneously.

`javax.tv.service.selection.InsufficientResourcesException` - If the operation cannot be completed due to a lack of system resources.

`java.lang.SecurityException` - If the caller does not have MediaSelectPermission(components[i]) permission for any valid i.

`java.lang.IllegalArgumentException` - if the two arrays are not the same size.

Since:

MHP 1.1.2.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 441 of 861

select(Locator, StreamType)

```
public void select(javax.tv.locator.Locator component, javax.tv.service.navigation.StreamType type)
throws InvalidLocatorException, InvalidServiceComponentException, InsufficientResourcesException, Se
curityException
```

Selects a new service component for presentation over-riding the stream type signalled for the component. If some content is currently playing, it is replaced in its entirety by the specified selection. This is an asynchronous operation that is completed upon receipt of a MediaSelectEvent. Note that for certain selections that imply a different time base or otherwise change synchronization relationships, a RestartingEvent will be posted by the Player.

Parameters:

`component` - A locator representing an individual service component to present.

`type` - The stream type of the component.

Throws:

`javax.tv.locator.InvalidLocatorException` - If the locator does not reference a selectable service component.

`javax.tv.service.selection.InvalidServiceComponentException` - If the specified service component is not part of the Service to which the MediaSelectControl is restricted, or if it cannot be presented alone.

`javax.tv.service.selection.InsufficientResourcesException` - If the operation cannot be completed due to a lack of system resources.

`java.lang.SecurityException` - If the caller does not have MediaSelectPermission(component) permission.

Since:

MHP 1.1.2.

selectServiceMediaComponents(Locator)

```
public void selectServiceMediaComponents(javax.tv.locator.Locator l)
throws InvalidLocatorException, InvalidServiceComponentException, InsufficientResourcesException
```

Selects for presentation the media service components from a service. If some content is currently playing, it is replaced in its entirety by the media service components from the specified service. This is an asynchronous operation that is completed upon receipt of a MediaSelectEvent. Note that for most selections that imply a different time base or otherwise change synchronization relationships, a RestartingEvent will be posted by the Player. The rules for deciding which media service components shall be presented are defined in the main body of the present document.

Parameters:

`l` - the locator for a service.

Throws:

`javax.tv.locator.InvalidLocatorException` - If the locator provided does not reference a service.

`javax.tv.service.selection.InvalidServiceComponentException` - If the locator provided does not reference a service which contains at least one media service component

`javax.tv.service.selection.InsufficientResourcesException` - If the operation cannot be completed due to a lack of system resources.

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

org.dvb.media

NoComponentSelectedEvent

Declaration

```
public class NoComponentSelectedEvent extends javax.media.StopEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--javax.media.ControllerEvent
              |
              +--javax.media.TransitionEvent
                    |
                    +--javax.media.StopEvent
                          |
                          +--org.dvb.media.NoComponentSelectedEvent
```

All Implemented Interfaces:

`javax.media.MediaEvent`, `java.io.Serializable`

Description

This event is generated whenever presentation of a stream stops because there are no selected components to present. One example of this would be use of the `javax.tv.media.MediaSelectControl.remove` method to remove all components of a service. Generation of this event informs the application that the Player is no longer presenting any content.

Since:

MHP 1.0.1.

Constructors

NoComponentSelectedEvent(Controller, int, int, int, MediaLocator)

```
public NoComponentSelectedEvent(javax.media.Controller source, int previous, int current,
int target, javax.media.MediaLocator stream)
```

Construct an event.

Parameters:

`source` - the controller which was presenting the service.

`stream` - the locator of the stream whose presentation has stopped.

`previous` - the previous state of the controller.

`current` - the current state of the controller.

`target` - the target state of the controller.

Methods

getStream()

```
public javax.media.MediaLocator getStream()
```

This method returns the stream whose presentation has stopped

Returns:

the locator for the stream concerned.

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

org.dvb.media
PresentationChangedEvent

Declaration

```
public class PresentationChangedEvent extends javax.media.ControllerEvent

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--javax.media.ControllerEvent
              |
              +--org.dvb.media.PresentationChangedEvent
```

All Implemented Interfaces:

`javax.media.MediaEvent, java.io.Serializable`

Description

This event is generated whenever the content being presented by a player changes for reasons outside the control of the application. The state of the player does not change - only the content being presented.

Fields

CA_FAILURE

```
public static final int CA_FAILURE
```

Presentation changed due an action by the CA subsystem. Alternate content is being played, not the content selected by the user (e.g. adverts in place of a scrambled service).

See Also:

`getReason()`

CA_RETURNED

```
public static final int CA_RETURNED
```

Presentation changed due to an action by the CA subsystem. Normal content is now being presented as requested by the user. This reason code is used when the CA subsystem commands the GEM terminal to switch back to the normal presentation after having previously selected an alternate content.

See Also:

`getReason()`

STREAM_UNAVAILABLE

```
public static final int STREAM_UNAVAILABLE
```

The stream being presented is no longer available in the transport stream.

See Also:

`getReason()`

Constructors

PresentationChangedEvent(Controller, MediaLocator, int)

```
public PresentationChangedEvent(javax.media.Controller source, javax.media.MediaLocator stream,
int reason)
```

Constructor for the event.

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

Parameters:

`source` - the controller whose presentation changed.

`stream` - the stream now being presented.

`reason` - the reason for the change encoded as one of the constants in this class.

Methods

getReason()

```
public int getReason()
```

This method returns the reason why access has been withdrawn.

Returns:

the reason for the change specified when the event was constructed.

getStream()

```
public javax.media.MediaLocator getStream()
```

This method returns the locator for the stream now being presented.

Returns:

the locator for the stream now being presented.

org.dvb.media

ServiceRemovedEvent

Declaration

```
public class ServiceRemovedEvent extends javax.media.StopEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--javax.media.ControllerEvent
              |
              +--javax.media.TransitionEvent
                    |
                    +--javax.media.StopEvent
                          |
                          +--org.dvb.media.ServiceRemovedEvent
```

All Implemented Interfaces:

`javax.media.MediaEvent`, `java.io.Serializable`

Description

This event is generated whenever access to a service stops because the service concerned has been removed from the network. Generation of this event informs the application that the Player is no longer presenting any content.

Since:

MHP 1.0.1.

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

Constructors

ServiceRemovedEvent(Controller)

```
public ServiceRemovedEvent(javax.media.Controller source)
```

Construct an event.

Parameters:

`source` - the controller which was presenting the service.

ServiceRemovedEvent(Controller, int, int, int, MediaLocator)

```
public ServiceRemovedEvent(javax.media.Controller source, int previous, int current, int target,
javax.media.MediaLocator stream)
```

Construct an event.

Parameters:

`source` - the controller which was presenting the service.

`stream` - the locator of the stream which was removed from the network.

`previous` - the previous state of the controller.

`current` - the current state of the controller.

`target` - the target state of the controller.

Methods

getStream()

```
public javax.media.MediaLocator getStream()
```

This method returns the stream which was removed from the network.

Returns:

the stream concerned.

org.dvb.media

StopByResourceLossEvent

Declaration

```
public class StopByResourceLossEvent extends javax.media.StopEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--javax.media.ControllerEvent
              |
              +--javax.media.TransitionEvent
                    |
                    +--javax.media.StopEvent
                          |
                          +--org.dvb.media.StopByResourceLossEvent
```

All Implemented Interfaces:

```
javax.media.MediaEvent, java.io.Serializable
```


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

Description

This event is generated whenever presentation of a stream stops because the player has lost so many resources that it cannot continue. Generation of this event informs the application that the Player is no longer presenting any content.

Since:

MHP 1.0.1.

Constructors

StopByResourceLossEvent(Controller, int, int, int, MediaLocator)

```
public StopByResourceLossEvent(javax.media.Controller source, int previous, int current, int target,
javax.media.MediaLocator stream)
```

Construct an event.

Parameters:

`source` - the controller which was presenting the service.

`stream` - the locator of the stream which was being presented.

`previous` - the previous state of the controller.

`current` - the current state of the controller.

`target` - the target state of the controller.

Methods

getStream()

```
public javax.media.MediaLocator getStream()
```

This method returns the stream which was being presented.

Returns:

the locator for the stream concerned.

org.dvb.media

SubtitleAvailableEvent

Declaration

```
public class SubtitleAvailableEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.media.SubtitleAvailableEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Report that subtitles are available to be presented having been unavailable. This event is not generated on service selection or other forms of 'zapping'. Its generation is restricted to changes in the composition of the subtitle aspects of the same broadcast stream.

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

Constructors

SubtitleAvailableEvent(Object)

```
public SubtitleAvailableEvent(java.lang.Object source)
```

Constructor.

Parameters:

`source` - the source of the event. The platform shall always pass in the JMF player presenting the subtitles.

Methods

getSource()

```
public java.lang.Object getSource()
```

Return the JMF player which is the source of the event.

Overrides:

`getSource` in class `EventObject`.

Returns:

the source of the event. This shall be the JMF player passed in to the constructor of the event.

org.dvb.media

SubtitleListener

Declaration

```
public interface SubtitleListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

Report that a subtitle event has happened.

Methods

subtitleStatusChanged(EventObject)

```
public void subtitleStatusChanged(java.util.EventObject event)
```

Report a subtitle event has happened.

Parameters:

`event` - the event which happened.

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

org.dvb.media SubtitleNotAvailableEvent

Declaration

```
public class SubtitleNotAvailableEvent extends java.util.EventObject

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.media.SubtitleNotAvailableEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Inform an application that a subtitle stream has vanished from the network. This event is not generated on service selection or other forms of 'zapping'. Its generation is restricted to changes in the composition of the subtitle aspects of the same broadcast stream.

Constructors

SubtitleNotAvailableEvent(Object)

```
public SubtitleNotAvailableEvent(java.lang.Object source)
```

Constructor.

Parameters:

`source` - the source of the event. The platform shall always pass in the JMF player presenting the subtitles.

Methods

getSource()

```
public java.lang.Object getSource()
```

Return the source of the event.

Overrides:

`getSource` in class `EventObject`.

Returns:

the source of the event. This shall be the JMF player passed in to the constructor of the event.

org.dvb.media SubtitleNotSelectedEvent

Declaration

```
public class SubtitleNotSelectedEvent extends java.util.EventObject

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.media.SubtitleNotSelectedEvent
```


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

All Implemented Interfaces:

`java.io.Serializable`

Description

Report that subtitles are not now selected. Even if subtitles are available in the network, they will not be presented. This event is generated when the combination of end user control of subtitles through the navigator and application control of subtitles through `SubtitlingLanguageControl.setSubtitling` changes whether subtitles are to be presented if they are available. It is not generated for changes in the underlying availability of subtitles even if those cause changes in whether subtitles are presented or not.

Constructors

SubtitleNotSelectedEvent(Object)

```
public SubtitleNotSelectedEvent(java.lang.Object source)
```

Constructor.

Parameters:

`source` - the source of the event. The platform shall always pass in the JMF player presenting the subtitles.

Methods

getSource()

```
public java.lang.Object getSource()
```

Return the source of the event

Overrides:

`getSource in class EventObject`

Returns:

the source of the event. This shall be the JMF player passed in to the constructor of the event.

org.dvb.media

SubtitleSelectedEvent

Declaration

```
public class SubtitleSelectedEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.media.SubtitleSelectedEvent
```

All Implemented Interfaces:

`java.io.Serializable`

Description

Report that subtitles are now selected. If subtitles are also available then they will be presented. This event is generated when the combination of end user control of subtitles through the navigator and application control of subtitles through `SubtitlingLanguageControl.setSubtitling` changes whether subtitles are to be presented if they are available. It is not generated for changes in the underlying availability of subtitles even if those cause changes in whether subtitles are presented or not.

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

Constructors

SubtitleSelectedEvent(Object)

```
public SubtitleSelectedEvent(java.lang.Object source)
```

Constructor.

Parameters:

`source` - the source of the event. The platform shall always pass in the JMF player presenting the subtitles.

Methods

getSource()

```
public java.lang.Object getSource()
```

Return the source of the event.

Overrides:

`getSource in class EventObject`

Returns:

the source of the event. This shall be the JMF player passed in to the constructor of the event.

org.dvb.media

SubtitlingEventControl

Declaration

```
public interface SubtitlingEventControl extends org.davic.media.SubtitlingLanguageControl
```

All Superinterfaces:

`javax.media.Control, org.davic.media.LanguageControl, org.davic.media.SubtitlingLanguageControl`

Description

Allow applications to register and unregister their interest in events related to the availability and presentation of subtitles.

Methods

addSubtitleListener(SubtitleListener)

```
public void addSubtitleListener(org.dvb.media.SubtitleListener l)
```

Add a listener for subtitle events.

Parameters:

`l` - the listener to report the events to.

removeSubtitleListener(SubtitleListener)

```
public void removeSubtitleListener(org.dvb.media.SubtitleListener l)
```

Remove a listener for subtitle events.

Parameters:

`l` - the listener to remove.

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

org.dvb.media

VideoFormatControl

Declaration

```
public interface VideoFormatControl extends javax.media.Control
```

All Superinterfaces:

```
javax.media.Control
```

Description

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

It is important to note that due to different video and display formats (and user preferences), not all of the full video frame may be displayed. Similarly, it may not always be possible to map video and graphics with perfect accuracy.

Signalling for the active format description shall be supported as defined in annex B of TS 101 154 [2].

Fields

AFD_14_9

```
public static final int AFD_14_9
```

Constant representing an MPEG active format description of 14:9 (centre).

AFD_14_9_TOP

```
public static final int AFD_14_9_TOP
```

Constant representing an MPEG active format description of 14:9 (top).

AFD_16_9

```
public static final int AFD_16_9
```

Constant representing an MPEG active format description of 16:9 (centre).

AFD_16_9_SP_14_9

```
public static final int AFD_16_9_SP_14_9
```

Constant representing an MPEG active format description of 16:9 (with shoot and protect 14:9 centre).

AFD_16_9_SP_4_3

```
public static final int AFD_16_9_SP_4_3
```

Constant representing an MPEG active format description of 16:9 (with shoot and protect 4:3 centre).

AFD_16_9_TOP

```
public static final int AFD_16_9_TOP
```

Constant representing an MPEG active format description of 16:9 (top).

AFD_4_3

```
public static final int AFD_4_3
```

Constant representing an MPEG active format description of 4:3 (centre).

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

AFD_4_3_SP_14_9

```
public static final int AFD_4_3_SP_14_9
```

Constant representing an MPEG active format description of 4:3 (with shoot & protect 14:9 centre).

AFD_GT_16_9

```
public static final int AFD_GT_16_9
```

Constant representing an MPEG active format description of greater than 16:9 (centre).

AFD_NOT_PRESENT

```
public static final int AFD_NOT_PRESENT
```

Constant showing an MPEG active format description is not present.

AFD_SAME

```
public static final int AFD_SAME
```

Constant representing an MPEG active format description that is the same as the coded frame.

ASPECT_RATIO_16_9

```
public static final int ASPECT_RATIO_16_9
```

Constant representing an aspect ratio of 16:9.

ASPECT_RATIO_2_21_1

```
public static final int ASPECT_RATIO_2_21_1
```

Constant representing an aspect ratio of 2.21:1.

ASPECT_RATIO_4_3

```
public static final int ASPECT_RATIO_4_3
```

Constant representing an aspect ratio of 4:3.

ASPECT_RATIO_UNKNOWN

```
public static final int ASPECT_RATIO_UNKNOWN
```

Constant representing an unknown aspect ratio.

DAR_16_9

```
public static final int DAR_16_9
```

Constant representing a display aspect ratio of 16:9.

DAR_4_3

```
public static final int DAR_4_3
```

Constant representing a display aspect ratio of 4:3.

DFC_PLATFORM

```
public static final int DFC_PLATFORM
```

Control over the decoder format conversions is returned to being managed by the platform. This is the same as the value used if no GEM application has set a video transformation. It is not required to correspond to a single decoder format conversion and may change over time as the video input format and signalling change. This constant can only be used to set the decoder format conversion. Reading the decoder format conversion shall always return the DFC used at the time concerned.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 453 of 861

DFC_PROCESSING_16_9_ZOOM

```
public static final int DFC_PROCESSING_16_9_ZOOM
```

The central 16:9 letterbox area of the 4:3 720 x 576 input grid is expanded to fill the 16:9 output frame.

DFC_PROCESSING_CCO

```
public static final int DFC_PROCESSING_CCO
```

A 4:3 central part out of the 720 x 576 input 16:9 frame is transferred into a 720 x 576 4:3 output frame.

DFC_PROCESSING_FULL

```
public static final int DFC_PROCESSING_FULL
```

The full 720 x 576 frame is transferred (this may be either 4:3 or 16:9; part of this may be black, e.g. in the "pillar box" cases).

DFC_PROCESSING_LB_14_9

```
public static final int DFC_PROCESSING_LB_14_9
```

The 720 x 576 input grid is transferred into a 14:9 LB in a 4:3 frame.

DFC_PROCESSING_LB_16_9

```
public static final int DFC_PROCESSING_LB_16_9
```

The 720 x 576 input grid is transferred into a 16:9 letterbox in a 4:3 frame.

DFC_PROCESSING_LB_2_21_1_ON_16_9

```
public static final int DFC_PROCESSING_LB_2_21_1_ON_16_9
```

The 720 x 576 input grid is transferred into a 2.21:1 letterbox in a 16:9 frame.

DFC_PROCESSING_LB_2_21_1_ON_4_3

```
public static final int DFC_PROCESSING_LB_2_21_1_ON_4_3
```

The 720 x 576 input grid is transferred into a 2.21:1 letterbox in a 4:3 frame.

DFC_PROCESSING_NONE

```
public static final int DFC_PROCESSING_NONE
```

Decoder format conversion is inactive.

DFC_PROCESSING_PAN_SCAN

```
public static final int DFC_PROCESSING_PAN_SCAN
```

A 4:3 part out of the 720 x 576 input 16:9 or 2.21:1 frame is transferred into a 720 x 576 4:3 output frame. The horizontal position of this part is determined by pan&scan vectors from the MPEG video stream.

DFC_PROCESSING_UNKNOWN

```
public static final int DFC_PROCESSING_UNKNOWN
```

Constant representing an unknown format conversion being performed by the decoder.

Methods

addVideoFormatListener(VideoFormatListener)

```
public void addVideoFormatListener(org.dvb.media.VideoFormatListener l)
```

Add a listener for VideoFormatChangedEvents.

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

Parameters:

l - the listener to add.

getActiveFormatDefinition()

```
public int getActiveFormatDefinition()
```

Return the value of the active_format field of the MPEG Active Format Description of the video if it is transmitted (one of the constants AFD_* above). If this field is not available then AFD_NOT_PRESENT is returned. The constant values for the constants representing the active format description should be identical to the values specified in ETR154, annex B.

Returns:

the value of the active_format field of the MPEG Active Format Description of the video if it is transmitted. If this field is not available, or the video is not MPEG, then AFD_NOT_PRESENT is returned.

getAspectRatio()

```
public int getAspectRatio()
```

Return the aspect ratio of the video as it is transmitted. If the aspect ratio is not known, ASPECT_RATIO_UNKNOWN is returned.

Returns:

the aspect ratio of the video.

getDecoderFormatConversion()

```
public int getDecoderFormatConversion()
```

Return a value representing what format conversion is being done by the decoder in the platform (one of the constants DFC_* above). A receiver may implement only a subset of the available options. This decoder format conversion may be active or not depending upon the mode of operation.

Returns:

the decoder format conversion being performed or DFC_PROCESSING_UNKNOWN if this is not known.

getDisplayAspectRatio()

```
public int getDisplayAspectRatio()
```

Return the aspect ratio of the display device connected to this GEM decoder (one of the constants DAR_* above).

Returns:

the aspect ratio of the display device connected to the decoder.

getVideoTransformation(int)

```
public org.dvb.media.VideoTransformation getVideoTransformation(int dfc)
```

This method returns a VideoTransformation object that corresponds with the specified Decoder Format Conversion when applied to the currently selected video. If the specified Decoder Format Conversion is not supported for the currently selected video, then this method returns null.

Parameters:

dfc - the Decoder Format Conversion (one of the DFC_* constants specified in this interface).

Returns:

the video transformation, or null if the specified Decoder Format Conversion is not supported for the currently selected video.

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

isPlatform()

```
public boolean isPlatform()
```

Test if control over the decoder format conversions is being managed by the platform as defined by `DFC_PLATFORM`.

Returns:

true if control over the decoder format conversions is being managed by the platform, false otherwise.

See Also:

`DFC_PLATFORM`

removeVideoFormatListener(VideoFormatListener)

```
public void removeVideoFormatListener(org.dvb.media.VideoFormatListener l)
```

Remove a listener for VideoFormatChangedEvents.

Parameters:

`l` - the listener to remove.

org.dvb.media

VideoFormatEvent

Declaration

```
public abstract class VideoFormatEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.media.VideoFormatEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Direct Known Subclasses:

```
ActiveFormatDescriptionChangedEvent, AspectRatioChangedEvent, DFCChangedEvent
```

Description

The base class for all other events relating to changes in video format.

Constructors

VideoFormatEvent(Object)

```
public VideoFormatEvent(java.lang.Object source)
```

Constructor.

Parameters:

`source` - the source of the event. The platform shall always pass in the JMF Player presenting the video whose format changed.

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

org.dvb.media
VideoFormatListener

Declaration

```
public interface VideoFormatListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

The listener used to receive video format events.

Methods

receiveVideoFormatEvent(VideoFormatEvent)

```
public void receiveVideoFormatEvent(org.dvb.media.VideoFormatEvent anEvent)
```

receive a VideoFormatEvent.

Parameters:

`anEvent` - the VideoFormatEvent that has been received.

org.dvb.media
VideoPresentationControl

Declaration

```
public interface VideoPresentationControl extends javax.media.Control
```

All Superinterfaces:

```
javax.media.Control
```

All Known Subinterfaces:

```
BackgroundVideoPresentationControl
```

Description

A control to support setting and querying the video presentation.

Note: For a component-based player the scaling and positioning of the video is done by manipulating the corresponding AWT component. The VideoPresentationControl only allows for the setting of the clipping region.

Note: If the hardware supports the positioning of interlaced video on even lines only (when counting from 0), then a component-based player is allowed to position the top of the video one line below where it should be.

For a background player there is the BackgroundVideoPresentationControl that allows for the setting of the clipping region, the position and the scaling of the video in one atomic action.

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

Fields

POS_CAP_FULL

```
public static final byte POS_CAP_FULL
```

Constant representing that the video can be positioned anywhere on the screen, even if a part of the video is off screen as a result of that.

POS_CAP_FULL_EVEN_LINES

```
public static final byte POS_CAP_FULL_EVEN_LINES
```

n Constant representing that the video can be positioned anywhere on the screen, even if a part of the video is off screen as a result of that, with the restriction that the field order is respected. This implies that interlaced video can be positioned on even lines only (when counting from 0).

POS_CAP_FULL_EVEN_LINES_IF_ENTIRE_VIDEO_ON_SCREEN

```
public static final byte POS_CAP_FULL_EVEN_LINES_IF_ENTIRE_VIDEO_ON_SCREEN
```

Constant representing that the video can be positioned anywhere on screen as long as all the video is on screen, with the restriction that the field order is respected. This implies that interlaced video can be positioned on even lines only (when counting from 0).

POS_CAP_FULL_IF_ENTIRE_VIDEO_ON_SCREEN

```
public static final byte POS_CAP_FULL_IF_ENTIRE_VIDEO_ON_SCREEN
```

Constant representing that the video can be positioned anywhere on screen as long as all the video is on screen.

POS_CAP_OTHER

```
public static final byte POS_CAP_OTHER
```

Constant representing that the video positioning capability cannot be expressed by another POS_CAP_* constant.

Methods

getActiveVideoArea()

```
public org.havi.ui.HScreenRectangle getActiveVideoArea()
```

This method returns the size and location of the active video area. The active video area excludes any "bars" used for letterboxing or pillarboxing that the receiver knows about. Bars that are included in the broadcast stream and not signalled by active format descriptors are included in the active video area. The active video area may be larger/smaller than the screen, and may possibly be offset. The offsets will be negative if the origin of the active video area is above/left of the top, left corner of the screen. In case of pan&scan, the value returned may vary over time. This method only describes the relationship between the active video and the screen. It does not describe which portion of the screen is displaying the video.

NOTE: This method includes any video scaling.

Returns:

an HScreenRectangle representing the active video area in the normalized coordinate space.

getActiveVideoAreaOnScreen()

```
public org.havi.ui.HScreenRectangle getActiveVideoAreaOnScreen()
```

This method returns the size and location of the active video area on-screen. The active video area excludes any "bars" used for letterboxing or pillarboxing that the receiver knows about. Bars that are included in the broadcast stream and not signalled by active format descriptors are included in the active video area. The active video area on-screen may be smaller than the area of the screen, and may possibly be offset a positive amount. This method only describes the area on-screen where active video is being presented. It does not really describe which part of the video is being shown on-screen. This is especially true for pan&scan.

NOTE: This method includes any video scaling.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 458 of 861

Returns:

an HScreenRectangle representing the active video area on-screen in the normalized coordinate space.

getClipRegion()

```
public java.awt.Rectangle getClipRegion()
```

This method returns the area of the decoded video that will be displayed. If clipping is not supported, the dimensions of the bounding box will be the same as the displayed video. Note that when the GEM terminal is in `pan & scan mode`, the return value of this method will be out of date almost as soon as the method has returned.

Returns:

area of the decoded video that will be displayed. The coordinate space used to express the region is that of the decoded video after possible ETR154 up-sampling.

getHorizontalScalingFactors()

```
public float[] getHorizontalScalingFactors()
```

This method gives information about the supported discrete horizontal scaling factors in case arbitrary horizontal scaling is not supported.

Returns:

an array with the supported discrete horizontal scaling factors (including the scaling factor 1), sorted in ascending order. null is returned when arbitrary horizontal scaling is supported.

getInputVideoSize()

```
public java.awt.Dimension getInputVideoSize()
```

This method returns the dimensions of the video before any scaling has taken place (but after TS 101 154 [2] up-sampling). On 50 Hz standard definition systems this method always returns 720 x 576.

Returns:

the size of the decoded video before any scaling has taken place (but after TS 101 154 [2] up-sampling).

getPositioningCapability()

```
public byte getPositioningCapability()
```

This method gives information about how the video can be positioned on screen.

Returns:

the positioning capability for the currently selected video as one of the POS_CAP_* constants.

getTotalVideoArea()

```
public org.havi.ui.HScreenRectangle getTotalVideoArea()
```

This method returns a relative size and location of the total video area, including any "bars" used for letterboxing or pillarboxing that are included in the broadcast stream, but excluding any "bars" introduced as a result of video filtering. This may be larger or smaller than the size of the physical display device. This method only describes the relationship between the total video and the screen. It does not describe which portion of the screen is displaying the video.

NOTE: This method includes any video scaling.

Returns:

an HScreenRectangle representing the total video area in the normalized coordinate space.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 459 of 861

getTotalVideoAreaOnScreen()

```
public org.havi.ui.HScreenRectangle getTotalVideoAreaOnScreen()
```

This method returns a relative size and location of the total video area on-screen, including any "bars" used for letterboxing or pillarboxing that are included in the broadcast stream, but excluding any "bars" introduced as a result of video filtering. This method only describes the area on-screen where total video is being presented. This does not really describe which part of the video is being shown on-screen. This is especially true for pan&scan.

NOTE: This method includes any video scaling.

Returns:

an HScreenRectangle representing the total video area on-screen in the normalized coordinate space.

getVerticalScalingFactors()

```
public float[] getVerticalScalingFactors()
```

This method gives information about the supported discrete vertical scaling factors in case arbitrary vertical scaling is not supported.

Returns:

an array with the supported discrete vertical scaling factors (including the scaling factor 1), sorted in ascending order. null is returned when arbitrary vertical scaling is supported.

getVideoSize()

```
public java.awt.Dimension getVideoSize()
```

This method returns the size of the decoded video as it is being presented to the user. It takes scaling and clipping into account.

Returns:

the size of the decoded video as it is being presented to the user.

setClipRegion(Rectangle)

```
public java.awt.Rectangle setClipRegion(java.awt.Rectangle clipRect)
```

Set the region of the decoded video that will be displayed. If clipping is not supported, this method has no effect. If the bounding box extends beyond the decoded video, the results are implementation dependent. By default, the clipping region is set to the dimensions of the decoded video. This method returns the bounding box of the clipping region that was actually set. Implementations may approximate the requested rectangle if they have restrictions on video clipping.

If the player is a component-based player (as opposed to a background player), then the top left corner of the clip region will be aligned with the top left corner of the java.awt.Component returned by the method javax.media.Player.get-VisualComponent(). Hence changing the position of the clip region within the video moves the video with respect to the coordinate space used by java.awt.

Parameters:

`clipRect` - the bounding box of the clipping region. The coordinate space used to express the region is that of the decoded video after possible TS 101 154 [2] up-sampling.

Returns:

the set clipping region. If the requested clipping region is supported exactly, then the input parameter clipRect is returned, otherwise a newly created object will be returned.

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

supportsArbitraryHorizontalScaling()

```
public float[] supportsArbitraryHorizontalScaling()
```

This method gives information about whether arbitrary horizontal scaling is supported for the currently playing video. If arbitrary horizontal scaling is supported, then an array with two elements in returned. The first element returns the smallest allowed scaling factor (e.g. 0,5) and the second element returns the largest allowed scaling factor (e.g. 4). If arbitrary horizontal scaling is not supported, null is returned. In that case the method getHorizontalScalingFactors can be used to query which discrete scaling factors are supported.

Returns:

an array with the minimum and maximum allowed horizontal scaling factor, or null if arbitrary horizontal scaling is not supported.

supportsArbitraryVerticalScaling()

```
public float[] supportsArbitraryVerticalScaling()
```

This method gives information about whether arbitrary vertical scaling is supported for the currently playing video. If arbitrary vertical scaling is supported, then an array with two elements in returned. The first element returns the smallest allowed scaling factor (e.g. 0,5) and the second element returns the largest allowed scaling factor (e.g. 2). If arbitrary vertical scaling is not supported, null is returned. In that case the method getVerticalScalingFactors can be used to query which discrete scaling factors are supported.

Returns:

an array with the minimum and maximum allowed vertical scaling factor, or null if arbitrary vertical scaling is not supported.

supportsClipping()

```
public boolean supportsClipping()
```

Test if the decoder supports clipping.

Returns:

true if and only if the decoder supports clipping.

org.dvb.media

VideoTransformation

Declaration

```
public class VideoTransformation

java.lang.Object
  |
  +--org.dvb.media.VideoTransformation
```

Description

VideoTransformation objects express video transformations, i.e. the clipping, the horizontal and vertical scaling and the position of the video. All transformations are to be applied after possible TS 101 154 [2] up-sampling.

NOTE: Instances of VideoTransformation can represent pan and scan, but an application cannot create such instances itself. An application can get a VideoTransformation representing pan and scan, by calling the VideoFormatControl.getVideoTransformation() method with the pan and scan Decoder Format Conversion constant.

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

Constructors

VideoTransformation()

```
public VideoTransformation()
```

Creates a VideoTransformation object with default parameters. Clipping is disabled, both the horizontal and the vertical scaling factors are 1, and the video position is (0,0) in the normalised coordinate space.

VideoTransformation(Rectangle, float, float, HScreenPoint)

```
public VideoTransformation(java.awt.Rectangle clipRect, float horizontalScalingFactor,
float verticalScalingFactor, org.havi.ui.HScreenPoint location)
```

Creates a VideoTransformation object with the supplied parameters.

Parameters:

`clipRect` - the bounding box of the clipping region. The coordinate space used to express the region is that of the decoded video after possible ETR154 up-sampling. A non-null ClipRect enables clipping. A null ClipRect disables it.

`horizontalScalingFactor` - the horizontal scaling factor.

`verticalScalingFactor` - the vertical scaling factor.

`location` - the location of the video on the screen in the normalised coordinate space.

Methods

getClipRegion()

```
public java.awt.Rectangle getClipRegion()
```

Gets the clipping region.

Returns:

the bounding box of the clipping region. The coordinate space used to express the region is that of the decoded video after possible ETR154 up-sampling. null is returned if this video transformation represents pan and scan or if clipping is disabled.

getScalingFactors()

```
public float[] getScalingFactors()
```

Gets the horizontal and vertical scaling factors.

Returns:

an array with two elements. The first element contains the horizontal scaling factor, the second element the vertical scaling factor.

getVideoPosition()

```
public org.havi.ui.HScreenPoint getVideoPosition()
```

Returns the video position.

Returns:

the location of the video on the screen in the normalised coordinate space.

isPanAndScan()

```
public boolean isPanAndScan()
```

Returns whether this video transformation represents pan and scan.

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

Returns:

true is this video transformation represents pan and scan, false otherwise.

setClipRegion(Rectangle)

```
public void setClipRegion(java.awt.Rectangle clipRect)
```

Sets the clipping region.

If this video transformation represents pan and scan, then it will no longer represent pan and scan when this method is called. A non-null ClipRect enables clipping. A null ClipRect disables it.

Parameters:

`clipRect` - the bounding box of the clipping region. The coordinate space used to express the region is that of the decoded video after possible ETR154 up-sampling.

setScalingFactors(float, float)

```
public void setScalingFactors(float horizontalScalingFactor, float verticalScalingFactor)
```

Sets the horizontal and vertical scaling factors.

Parameters:

`horizontalScalingFactor` - the horizontal scaling factor.

`verticalScalingFactor` - the vertical scaling factor.

setVideoPosition(HScreenPoint)

```
public void setVideoPosition(org.havi.ui.HScreenPoint location)
```

Sets the video position.

Parameters:

`location` - the location of the video on the screen in the normalised coordinate space.

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

N.2 Streaming monitoring API

Package org.dvb.media.monitoring

Description

Provides DVB specific extensions to the Java Media Framework related to stream monitoring.

Class Summary	
Interfaces	
MonitoringControl	A control allows applications to get information associated with the Streamed CoD being presented to the user.
MonitoringListener	The listener used to receive MonitoringEvents.
Classes	
BufferingStatusEvent	Event signalling a change in the buffer status.
MonitoringEvent	The base class for all other events related to changes in associated CoD.
StreamQualityChangeEvent	Event signals that quality of the associated CoD has been changed.
StreamQualityInfo	Represents information about quality of a CoD stream.
VideoStreamQualityInfo	Represents information about CoD stream's video.

MonitoringControl

All Superinterfaces:

javax.media.Control

```
public interface MonitoringControl extends javax.media.Control
```

A MonitoringControl allows applications to get information associated with the Streamed CoD being presented to the user.

Additionally it allows applications to control quality of CoD by setting bitrate of downloaded CoD.

addMonitoringListener

```
void addMonitoringListener(MonitoringListener l)
```

Add a listener for MonitoringEvent.

Parameters:

l - the listener to add

getAudioFormat

```
java.lang.String getAudioFormat()
   throws java.lang.IllegalStateException
```

Get the audio format of the associated CoD.

Formats are encoded as MPEG-7 term IDs as extended by the DVB IPI handbook, e.g. MPEG-1 layer 3 audio is encoded as "3.3". Precognitions: Information is available after prefetching.

Returns:

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

audio format of the associated CoD

Throws:

java.lang.IllegalStateException - thrown if Player is neither in Prefetched nor Started state

getAvailableStreamQualities

```
StreamQualityInfo[] getAvailableStreamQualities()
   throws java.lang.IllegalStateException
```

Get all available qualities of streams in associated CoD. In case of video CoD an array of VideoStreamQualityInfos shall be returned. Precognitions: Information is available after prefetching

Returns:

all available qualities of associated CoD

Throws:

java.lang.IllegalStateException - if Player is neither in Prefetched nor in Started state

getCurrentLineRate

```
long getCurrentLineRate()
```

Get the current speed (in bits per second) with which the bits are sent onto the wire.

Returns:

current speed with which the bits are sent onto the wire

getStreamQuality

```
StreamQualityInfo getStreamQuality()
   throws java.lang.IllegalStateException
```

Get the current quality of associated CoD. It is the quality at which associated CoD is being downloaded and presented to the user. In case of video CoD an instance of VideoStreamQualityInfo shall be returned. Precognitions: Information is available after prefetching

Returns:

quality of stream at which associated CoD is being downloaded and presented to the user

Throws:

java.lang.IllegalStateException - if Player is neither in Prefetched nor Started state

getVideoFormat

```
java.lang.String getVideoFormat()
   throws java.lang.IllegalStateException
```

Get the video format of associated CoD.

Formats are encoded as MPEG-7 term IDs as extended by the DVB IPI handbook, e.g. MPEG-2 video, main profile, main level is encoded as "2.2.2". Precognitions: Information is available after prefetching

Returns:

video format of the associated CoD or null if CoD does not have video stream

Throws:

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

java.lang.IllegalStateException - thrown if Player is neither in Prefetched nor Started state

removeMonitoringListener

```
void removeMonitoringListener(MonitoringListener l)
```

Remove a listener for MonitoringEvent.

Parameters:

l - the listener to remove

setFixedStreamQuality

```
void setFixedStreamQuality(StreamQualityInfo streamQuality)
   throws   java.lang.IllegalArgumentException,
      java.lang.IllegalStateException
```

Set fixed quality of associated CoD. It allows application to force downloading CoD with a given video resolution and stream bitrate, and thus disable adaptive streaming support for associated CoD.

Application can check CoD's available qualities using getAvailableStreamQualities() method.

Parameters:

streamQuality - the stream quality of CoD

Throws:

java.lang.IllegalArgumentException - thrown if CoD does not support given quality

java.lang.IllegalStateException - if Player is not in the Prefetched state

setInitialStreamQuality

```
void setInitialStreamQuality(StreamQualityInfo streamQuality)
   throws   java.lang.IllegalArgumentException,
               java.lang.IllegalStateException
```

Set initial quality of associated CoD. After Player starts, a quality could be changed by Adaptive Streaming.

Application can check CoD's available qualities using getAvailableStreamQualities() method.

Parameters:

streamQuality - the stream quality of CoD

Throws:

java.lang.IllegalArgumentException - thrown if CoD does not support given quality

java.lang.IllegalStateException - if Player is not in the Prefetched state

StreamQualityInfo

```
java.lang.Object
  └ org.dvb.media.monitoring.StreamQualityInfo
```


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

Direct Known Subclasses:

VideoStreamQualityInfo

```
public class StreamQualityInfo extends java.lang.Object
```

Represents information about quality of a CoD stream. It contains information about stream bitrate (bits per second).

StreamQualityInfo

```
public StreamQualityInfo(long bitrate)
```

Construct a stream quality entity.

Parameters:

`bitrate` - stream bitrate

getStreamBitrate

```
public long getStreamBitrate()
```

Get stream's bitrate (bits per second) of a CoD.

Returns:

Bitrate of a CoD

VideoStreamQualityInfo

```
java.lang.Object
  └ org.dvb.media.monitoring.StreamQualityInfo
      └ org.dvb.media.monitoring.VideoStreamQualityInfo
```

```
public class VideoStreamQualityInfo extends StreamQualityInfo
```

Represent information about quality of a video CoD stream. It contains information about video resolution and stream bitrate (bits per second).

Instance of this class shall be returned by `MonitoringControl.getAvailableStreamQualities()` and `MonitoringControl.getStreamQuality()` in case when video stream is playing. Additionally, in this case an instance of this class shall be passed to `MonitoringControl.setFixedStreamQuality(StreamQualityInfo)` and `MonitoringControl.setFixedStreamQuality(StreamQualityInfo)` methods.

VideoStreamQualityInfo

```
public VideoStreamQualityInfo(java.awt.Dimension videoResolution, long bitrate)
```

Construct an stream quality entity.

Parameters:

`videoResolution` - video resolution

`bitrate` - stream bitrate in bits per second

getVideoResolution

```
public java.awt.Dimension getVideoResolution()
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 467 of 861

Get video resolution of a CoD.

Returns:

video resolution of a CoD

MonitoringListener

```
public interface MonitoringListener
```

The listener used to receive MonitoringEvent.

receiveMonitoringEvent

```
void receiveMonitoringEvent(MonitoringEvent anEvent)
```

Receive a MonitoringEvent.

Parameters:

`anEvent` - the MonitoringEvent that has been received

MonitoringEvent

```
java.lang.Object
  └ java.util.EventObject
      └ org.dvb.media.monitoring.MonitoringEvent
```

All Implemented Interfaces:

java.io.Serializable

Direct Known Subclasses:

BufferingStatusEvent, StreamQualityChangeEvent

```
public class MonitoringEvent extends java.util.EventObject
```

The base class for all other events related to changes in associated CoD.

MonitoringEvent

```
protected MonitoringEvent(java.lang.Object source)
```

Construct the event.

Parameters:

`source` - the source of the event. The platform shall always pass in the JMF Player presenting the media whose property has been changed

StreamQualityChangeEvent

```
java.lang.Object
```


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

```
└java.util.EventObject
   └org.dvb.media.monitoring.MonitoringEvent
      └org.dvb.media.monitoring.StreamQualityChangeEvent
```

All Implemented Interfaces:

java.io.Serializable

```
public class StreamQualityChangeEvent extends MonitoringEvent
```

Event is signalling that quality of the associated CoD has been changed.

This event shall occur when adaptive streaming changes played stream to another stream with a different video resolution and/or stream's bitrate because of changing network conditions.

StreamQualityChangeEvent

```
public StreamQualityChangeEvent(java.lang.Object source, StreamQualityInfo newStreamQuality)
```

Construct the event.

Parameters:

`source` - the source of the event. The platform shall always pass in the JMF Player presenting the media whose property has been changed

`newStreamQuality` - the new value of quality of the stream

getStreamQuality

```
public StreamQualityInfo getStreamQuality()
```

Get the new quality of associated CoD.

Returns:

the new stream's quality

BufferingStatusEvent

```
java.lang.Object
└java.util.EventObject
   └org.dvb.media.monitoring.MonitoringEvent
      └org.dvb.media.monitoring.BufferingStatusEvent
```

All Implemented Interfaces:

java.io.Serializable

```
public class BufferingStatusEvent extends MonitoringEvent
```

Event signalling a change in the buffer status.

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

Events are sent when a JMF Player is in the Started state.

To prevent implementations from being overloaded BUFFERING_STATUS_UNDERRUN and BUFFERING_STATUS_BACK_TO_NORMAL events should not be sent faster than one per seconds.

BUFFERING_STATUS_BACK_TO_NORMAL

```
public static final int BUFFERING_STATUS_BACK_TO_NORMAL
```

Indicates that buffering status back to a normal state from BUFFERING_STATUS_UNDERRUN or BUFFERING_STATUS_STOPPED states.

BUFFERING_STATUS_STOPPED

```
public static final int BUFFERING_STATUS_STOPPED
```

Indicates that buffering has been stopped. It means that network connection has been lost.

BUFFERING_STATUS_UNDERRUN

```
public static final int BUFFERING_STATUS_UNDERRUN
```

Indicates that buffering status is underrun. It means that coding rate is too high compared with network line rate.

BufferingStatusEvent

```
public BufferingStatusEvent(java.lang.Object source, int bufferingStatus)
   throws java.lang.IllegalArgumentException
```

Construct the event.

Parameters:

source - the source of the event. The platform shall always pass in the JMF Player presenting the media whose property has been changed

bufferingStatus - the status of the buffer

Throws:

java.lang.IllegalArgumentException - thrown if bufferingStatus is out of range

getBufferingStatus

```
public int getBufferingStatus()
```

Get buffering status.

Returns:

status of the buffer

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 470 of 861

N.3 Media Stream Synchronization API

Package

org.dvb.media

Description

Provides DVB specific extensions to the Java Media Framework to enable synchronization of media streams.

Class Summary	
Interfaces	
MasterSlaveSyncLinkageControl	MasterSlaveSyncLinkageControl handles Player-based Master/Slave relationship for synchronization of JMF Players.
SyncControl	SyncControl is a Control object for synchronization of a slave JMF Player.
SyncStateChangedEventListener	This listener interface is for receiving synchronization change events among JMF Players.
Classes	
SyncStateChangedEvent	An event which indicates change of status of synchronization between master/slave JMF Players.
Exceptions	
IncompatibleSynchronizableMediaTypeException	This exception is thrown when media type of each player is incompatible for synchronization.

MasterSlaveSyncLinkageControl

```
package org.dvb.media
public interface MasterSlaveSyncLinkageControl
```

MasterSlaveSyncLinkageControl handles Player-based Master/Slave relationship for synchronization of JMF Players. A JMF Player that provides master clock for synchronization is a *master Player*. MasterSlaveSyncLinkageControl allows a JMF Player from which this Control is obtained to be a master Player. At the time MasterSlaveSyncLinkageControl object is obtained from a Controller object of a JMF Player, the Player is a potential master Player. When a Controller object of a *slave Player* is added this control by `addSlave()` method, actual synchronization relationship is established. If synchronization between the Players is successfully established, a JMF Player to which MasterSlaveSyncLinkageControl object belongs becomes a master Player. When all the slave Players are removed from the MasterSlaveSyncLinkageControl object by `removeSlave()` method, a master Player is back to a potential master Player.

MasterSlaveSyncLinkageControl object monitors synchronization status between the master and slave Players. When synchronization status is changed, MasterSlaveSyncLinkageControl object notifies the situation by throwing SyncStateChangedEvent to the application. Before adding a slave Player, synchronization status is 'Unsynchronized'. At the time a slave Player is registered by `addSlave()` method, synchronization status is 'Synchronizing'. When synchronization is successfully established, the status is 'Synchronized'. In case that established synchronization cannot be kept within given tolerance, the status returns to 'Synchronizing' and related control objects try to re-synchronize automatically unless synchronization status is set to *OUT_OF_SYNC_UNSYNCHRONIZABLE*.

MasterSlaveSyncLinkageControl supports only simple Master/Slave relationship. That is, the object accepts relationship between a single master Player and slave Players. It is not allowed that a single JMF Player becomes a master Player and a slave Player at the same time. (A potential master Player can be a slave Player.) In addition, a slave Player can have only a single master Player. In case that an application tries to establish the relationship that violates these conditions, *IllegalArgumentException* is thrown when `addSlave()` is called.

Instance of MasterSlaveSyncLinkageControl can be obtained from a JMF Controller via the methods getControl(String) and getControls(). A single Controller object of a JMF Player can create at most one instance of MasterSlaveSyncLinkageControl.

Synchronization Status and Status Constants

Unsynchronized Master/Slave relationship is not established and Players can be running freely.	Synchronizing *MASTERPLAYER_STOOPED, SLAVEPLAYER_STOPPED, OUT_OF_SYNC_MASTER_CLK, OUT_OF_SYNC_SLAVE_CLK, OUT_OF_SYNC_OTHERSs*	Synchronized *IN_SYNC*
		Unsynchronizable *OUT_OF_SYNC_UNSYNCHRONIZABLE*

IN_SYNC

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

```
static final int IN_SYNC
```
Status representing synchronized

MASTERPLAYER_STOPPED

```
static final int MASTERPLAYER_STOPPED
```
Status representing stop() method is applied to a master Player

OUT_OF_SYNC_MASTER_CLK

```
static final int OUT_OF_SYNC_MASTER_CLK
```
Status representing out of sync due to unstable clock of master Player

OUT_OF_SYNC_OTHERS

```
static final int OUT_OF_SYNC_OTHERS
```
Status representing out of sync due to other reasons

OUT_OF_SYNC_SLAVE_CLK

```
static final int OUT_OF_SYNC_SLAVE_CLK
```
Status representing out of sync due to unstable clock of slave Player

OUT_OF_SYNC_UNSYNCHRONIZABLE

```
static final int OUT_OF_SYNC_UNSYNCHRONIZABLE
```
Status representing out of sync and Unsynchronzable state

SLAVEPLAYER_STOPPED

```
static final int SLAVEPLAYER_STOPPED
```
Status representing stop() method is applied to a slave Player

addSlave

```
void addSlave(Controller ctrl)
         throws    InsufficientResourcesException,
                   IncompatibleSynchronizableMediaTypeException,
                   java.lang.IllegalArgumentException
```

Adds a specified Player as a slave Player that synchronizes with a master Player. Offset value between master Player and a specified Player is implicitly given as zero in this method. If the specified Player already establishes synchronization, this method does nothing.

Parameters:

`ctrl` - Controller object of a potential slave Player to synchronize with this master Player.

Throws:

`InsufficientResourcesException` - If the operation cannot be completed due to a lack of system resources.

`IncompatibleSynchronizableMediaTypeException` - If media type Player handles is incompatible to establish synchronization.

`java.lang.IllegalArgumentException` - If a specified Controller object already forms a master or slave Player.

addSlave

```
void addSlave(Controller ctrl, Time t)
         throws    InsufficientResourcesException,
                   IncompatibleSynchronizableMediaTypeException,
                   java.lang.IllegalArgumentException
```

Adds a specified Player as a slave Player that synchronizes with a master Player. Offset value given as a Time object states offset of time position of the specified slave Player and the master Player. If the specified Player already establishes synchronization, this method does nothing.

Parameters:

`ctrl` - Controller object of a potential slave Player to synchronize with this master Player.

`t` - Time object to represent offset between master Player's clock and the specified Player.

Throws:

`InsufficientResourcesException` - If the operation cannot be completed due to a lack of system resources.

`IncompatibleSynchronizableMediaTypeException` - If media type Player handles is incompatible to establish synchronization.

`java.lang.IllegalArgumentException` - If a specified Controller object already forms a master or slave Player.

addSyncStateChangedEventListener

```
void addSyncStateChangedEventListener(SyncStateChangedEventListener l)
```
Adds the specified SyncStateChangedEventListener to receive the events. SyncStateChangedEvent occurs when synchronization status between the master Player and any slave Player is changed. If l is null, no exception is thrown and no action is performed.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 472 of 861

Parameters:
l - SyncStateChangedEvent listener.

getSlaves

```
Controller[] getSlaves()
```
Gets slave Players added by addSlave method. Actual synchronization status of each slave Player is not considered in return value of this method.
Returns:
An array of Controller objects which are added by addSlave method.

getSyncStateChangedEventListeners

```
SyncStateChangedEventListener[] getSyncStateChangedEventListeners()
```
Returns an array of all the SyncStateChangedEvent listeners registered on this control.
Returns:
An array of registered SyncStateChangedEvent listeners.

getSyncStatus

```
int getSyncStatus(Controller ctrl)
```
Gets currrent synchronization status of s specified slave Player.
Parameters:
ctrl - Controller objcet of a slave Player object to obtain current synchronization status.
Returns:
Synchronization status in field values.

removeSlave

```
void removeSlave(Controller ctrl)
             throws java.lang.IllegalArgumentException
```
Removes a specified slave Player. The specified Player turns self-running. This removal should not affect any progress of a master Player.
Parameters:
ctrl - Controller object of a potential slave Player to synchronize with this master Player.
Throws:
java.lang.IllegalArgumentException - If the specified Controller object is not registered as a slave Player.

removeSyncStateChangedEventListener

```
void removeSyncStateChangedEventListener(SyncStateChangedEventListener l)
```
Removes a specified SyncStateChangedEventListener to receive the events. If l is null, no exception is thrown and no action is performed.
Parameters:
l - SyncStateChangedEvent listener.

SyncControl

```
package org.dvb.media
public interface SyncControl
```

SyncControl is a Control object for synchronization of a slave JMF Player. SyncControl manipulates offset in time with a master Player and allowable torelance of synchronization. When SyncControl object is obtained from a Controller object of a JMF Player by getControl(String) or getControls() methods, the JMF Player is a potential slave Player. Such a JMF Player becomes a slave Player when the Player is registered as a slave Player by `addSlave()` methods of MasterSlaveSyncLinkageControl.

SyncControl supports only simple Master/Slave relationship. That is, the object accepts relationship between a single master Player and slave Players. It is not allowed that a single JMF Player becomes a master Player and a slave Player at the same time. In addition, a slave Player can have only a single master Player.

Instance of SyncControl can be obtained from a JMF Player via the methods getControl(String) and getControls(). A single JMF Player can create at most one instance of SyncControl.

getMaster

```
MasterSlaveSyncLinkageControl getMaster()
```
Gets a MasterSlaveSinkLinkageControl object that this SyncControl is currently synchronizing with.
Returns:
MasterSlaveSinkLinkageControl object that this SyncControl is currently synchronizing with.

getOffset

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

`Time` **`getOffset`**`()`
Gets offset value of a specified slave Player against master Player, by Time object.
Returns:
Offset value in terms of Time object.

getTolerance

`Time` **`getTolerance`**`()`
Gets assigned tolerance value of synchronization for a specified slave Player.
Returns:
Value of tolerance for synchronization in terms of Time object.

setOffset

`void` **`setOffset`**`(Time t)`
Sets offset value in time for a slave Player to the master Player. The value is given by Time object.
Parameters:
`t` - Offset value in terms of Time object.

setTolerance

`void` **`setTolerance`**`(Time t)`
Sets threshold to determine whether or not a specified slave Player establishes synchronization. In other words, an application allows error of synchronization within a given parameter, t.
Parameters:
`t` - Allowable tolerance of synchronization in terms of Time object.

SyncStateChangedEvent

```
java.lang.Object
  └ java.util.EventObject
      └ org.dvb.media.SyncStateChangedEvent
```

All Implemented Interfaces:
java.io.Serializable

```
package org.dvb.media
public class SyncStateChangedEvent extends java.util.EventObject
```

An event which indicates change of status of synchronization between master/slave JMF Players.

This event is generated by MasterSlaveSyncLinkageControl object when status of synchronization between any combination of the master Player and slave Players.

SyncStateChangedEvent

`public` **`SyncStateChangedEvent`**`(java.lang.Object source, Controller ctrl,int status)`
Constructs a SyncStateChangedEvent object.
Parameters:
`ctrl` - Controller object of a slave Player that is related to this object.
`status` - current status of synchronization as a result of its change.

getSourceController

`public Controller` **`getSourceController`**`()`
Gets Controller object of a slave JMF Player object that is related to this event.
Returns:
A slave Player object related to this event object.

getSyncStatus

`public int` **`getSyncStatus`**`()`
Gets changed synchronization status.
Returns:
changed synchronization status.

SyncStateChangedEventListener

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

All Superinterfaces:
java.util.EventListener

```
package org.dvb.media
public interface SyncStateChangedEventListener extends java.util.EventListener
```

This listener interface is for receiving synchronization change events among JMF Players. The class that is interested in management of synchronization among JMF Players is registered with MasterSlaveSyncLinkageControl object using addSyncStateChangedEventListener method. The event is generated when synchronization status among JMF Players is changed. The relevant method in the listener object is then invoked, and the SyncStateChangedEvent is passed to it.

inSync

```
void inSync(SyncStateChangedEvent e)
```
Invoked when any combination of master and slave Players are synchronized.

outOfSync

```
void outOfSync(SyncStateChangedEvent e)
```
Invoked when any combination of master and slave Players are desynchronized.

unSynchronizable

```
void unSynchronizable(SyncStateChangedEvent e)
```
Invoked when synchronization is never established.

IncompatibleSynchronizableMediaTypeException

```
java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ org.dvb.media.IncompatibleSynchronizableMediaTypeException
```

All Implemented Interfaces:

java.io.Serializable

```
package org.dvb.media
public class IncompatibleSynchronizableMediaTypeException extends java.lang.Exception
```

This exception is thrown when media type of each player is incompatible for synchronization.

IncompatibleSynchronizableMediaTypeException

```
public IncompatibleSynchronizableMediaTypeException()
```

Constructs IncompatibleSynchronizableMediaTypeException.

IncompatibleSynchronizableMediaTypeException

```
public IncompatibleSynchronizableMediaTypeException(java.lang.String reason)
```

Constructs IncompatibleSynchronizableMediaTypeException with reason.

Parameters:

`reason` - String object describing the reason of this exception.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 475 of 861

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

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

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 476 of 861

Annex P (normative): Broadcast transport protocol access

P.1 Overview

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

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

a) the nature of a broadcast filesystem and its latency (e.g. possibility to asynchronously load the objects);

b) the type of the objects that can be encapsulated in a carousel and that do not exist in a classical file structure. These are: ServiceGateway, Stream and StreamEvent.

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

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

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

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

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

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

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

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

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

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

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

P.2 The org.dvb.dsmcc package

P.2.0 General

This package is derived from the `org.dvb.dsmcc` package as defined in clause P.4. In GEM these methods are bound to the more abstract signalling requirements of annex B.

The description of each class from the org.dvb.dsmcc package defined in clause P.4 is included in the present document, except as modified below.

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

P.2.1 DSMCCObject

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

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

P.2.1.1 DSMCCObject.getSigners()

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

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

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

P.2.1.2 DSMCCObject.getSigners(boolean known_root)

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

P.2.2 DSMCCStream

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

P.2.2.1 isAudio() method

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

P.2.2.2 isData() method

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

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

P.2.2.3 isMPEGProgram() method

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

P.2.2.4 isVideo() method

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

P.2.3 DSMCCStreamEvent

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

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

P.2.3.1 Lightweight binding of trigger API

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

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

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

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

P.2.3.1.1 DSMCCStreamEvent.getEventList()

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

P.2.3.1.2 StreamEvent.getEventId()

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

P.2.3.1.3 DSMCCStreamEvent.unsubscribe(int, StreamEventListener)

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

P.2.4 InvalidFormatException

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

P.2.5 ServiceDomain

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

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

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

P.2.5.1 ServiceDomain.attach(byte[])

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

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

P.2.5.2 ServiceDomain.attach

P.2.5.2.1 ServiceDomain.attach(Locator)

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

P.2.5.2.2 ServiceDomain.attach(Locator, int)

The locator parameter is to be interpreted as any locator that refers to a service. The integer is to be interpreted as a unique identification of a service domain within that service. Locator formats are discussed in clause 14.8. The method `ServiceDomain.attach(Locator,int)` is not applicable in IPTV and shall always fail with a ServiceXFRException.

NOTE: This method is intended for referencing multiple carousels in-band in a single service, e.g. separate carousels for applications and for locally inserted content. This concept is really only applicable in uni-directional networks.

P.2.5.3 ServiceDomain.getLocator()

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

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

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

P.2.5.4 ServiceDomain.getNSAPAddress()

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

P.2.5.5 ServiceDomain.getURL(Locator)

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

Returns a URL corresponding to a locator referring to a file or a directory, as specified in table 59 in clause 14.8, "Locators and content referencing." If the service domain corresponding to the locator is attached and the file or directory referenced in the locator exists then an instance of java.net.URL is returned which can be used to reference this object.

Parameters:

`l` - a locator referring to a file or directory, as specified in table 59.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 480 of 861

Returns:

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

Throws:

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

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

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

P.2.5.6 ServiceDomain.isNetworkConnectionAvailable()

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

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

P.2.6 ServiceXFRErrorEvent

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

P.2.7 ServiceXFRException

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

P.2.8 ServiceXFRReference

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

P.2.9 StreamEvent

Throughout this class, references to the DSMCC stream event descriptor are to be read as referring to the trigger event, as described in clause B.4.2. References to the event data refer to the payload defined in that clause. References to "NPT" and "eventNPT" refer to "timebase values" and "timebase_value", respectively.

P.3 Support for Stored Applications

The API defined in this clause allows DVB-J applications direct access to information broadcast according to annex B of the present document. GEM terminal specifications may make other filesystems available via this API (e.g. for files stored in local storage within the GEM terminal). These other file systems may subset the semantics of this package as follows:

- `DSMCCObject.abort()` - shall always throw a `NothingToAbortException`;
- `DSMCCObject.asynchronousLoad(AsynchronousLoadingEventListener)` - if the file exists, succeeds immediately (with `SuccessEvent` being generated) otherwise fails with `InvalidPathNameException`;
- `DSMCCObject.isObjectKindKnown()` - always returns true;
- `DSMCCObject.isStream()` - always returns false;

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

- `DSMCCObject.isStreamEvent()` - always returns false;
- `DSMCCObject.loadDirectoryEntry(AsynchronousLoadingEventListener)` - always succeeds immediately with `SuccessEvent` being generated;
- `DSMCCObject.prefetch(both signatures)` - always returns false;
- `DSMCCObject.setRetrievalMode(int)` - silently ignored;
- `DSMCCObject.synchronousLoad()` - if the file exists, succeeds immediately otherwise fails with `InvalidPathnameException`.

Additionally

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

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

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

P.4 Javadoc for org.dvb.dsmcc package

Package org.dvb.dsmcc

Description

Provides extended access to files carried in the broadcast stream. It includes some extensions to java.io which are generic to (possibly) long-latency file systems and some concepts which are specific to the DSMCC object carousel.

Class Summary	
Interfaces	
`AsynchronousLoadingEventListener`	Listener for applications which perform asynchronous loading, in order to be informed if the loading is done or if an error has occurred.
`NPTListener`	Objects that implement the `NPTListener` interface can receive `NPTStatusEvent` and `NPTRateChangedEvent` events.
`ObjectChangeEventListener`	The objects that implements the ObjectChangeEventListener interface can receive ObjectChangeEvent event.
`StreamEventListener`	Objects that implement the StreamEventListener interface can receive StreamEvent event.
Classes	
`AsynchronousLoadingEvent`	This class described an Object event which is used to notify the loading of a DSMCC object.
`DSMCCObject`	A `DSMCCObject` extends the java.io.File class with a combination of features appropriate for any file system with a long access time and features specific to the DSMCC object carousel as one example of such a file system.
`DSMCCStream`	The DSMCCStream class is used to manage DSMCC Stream Objects.
`DSMCCStreamEvent`	The DSMCCStreamEvent class is used to manage DSMCC StreamEvent Objects.
`InsufficientResourcesEvent`	This event is generated if there are insufficient resources available to load a DSMCCObject. e.g. if there is not enough memory.
`InvalidFormatEvent`	This event is generated if the format of the data received is inconsistent.

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

Class Summary	
InvalidPathnameEvent	The pathname does not exist or the ServiceDomain has been detached.
LoadingAbortedEvent	This event will be sent to the AsynchronousEventListener when an asynchronous loading operation is aborted.
MPEGDeliveryErrorEvent	An MPEGDeliveryErrorEvent indicates that an error (for instance, a time out or accessing the data would require tuning) has occurred while loading data from an MPEG Stream.
NotEntitledEvent	This event is sent when an attempt to asynchronously load an object has failed because the elementary stream carrying the object is scrambled and the user is not entitled to access the content of the object.
NPTDiscontinuityEvent	Sent when an MHP terminal detects a permanent value discontinuity in a broadcast timebase as defined in the main body of the present document.
NPTPresentEvent	Sent to listeners on a DSMCCStream object when a broadcast timeline newly appears for that DSMCC stream when it was not previously present.
NPTRate	Represents the rate at which an broadcast timeline progresses.
NPTRateChangeEvent	Sent only when the rate of an broadcast timeline changes value.
NPTRemovedEvent	Sent to listeners on a DSMCCStream object when a broadcast timeline which was present for that DSMCC stream is removed.
NPTStatusEvent	Sent when an MHP terminal detects a change of status in the broadcast timeline of a stream.
ObjectChangeEvent	This class describes an object change event that is used to monitor the arrival of a new version of a `DSMCCObject`.
ServerDeliveryErrorEvent	The local machine can not communicate with the server.
ServiceDomain	A `ServiceDomain` represents a group of DSMCC objects.
ServiceXFRErrorEvent	The object requested is available in an alternate ServiceDomain.
ServiceXFRReference	A ServiceXFRReference object is used when a DSMCC Object can not be loaded in the current ServiceDomain but is available in an alternate Service-Domain.
StreamEvent	This class describes a Stream event which is used to synchronize an application with an MPEG Stream.
SuccessEvent	This event indicates that the asynchronous loading was successful.
Exceptions	
DSMCCException	The DSMCCException is the root class of all DSMCC related exceptions
IllegalObjectTypeException	This Exception is thrown when the application attempted to create a DSMCCStream or DSMCCStreamEvent object with an object or a path that did not correspond to a DSMCC Stream or DSMCC StreamEvent respectively
InsufficientResourcesException	This exception gets thrown when a request to perform a DSMCC related operation cannot be completed due to resource limitations.
InvalidAddressException	An InvalidAddressException is thrown when the format of an NSAP address is not recognized.
InvalidFormatException	An InvalidFormatException is thrown when an inconsistent DSMCC message is received.
InvalidPathNameException	The InvalidPathNameException is thrown when the path name to a DSMCCObject does not exist or if the service domain is not detached.
MPEGDeliveryException	An MPEGDEliveryException is thrown when an error (for instance, a time out or accessing the data would require tuning) occurs while loading data from an MPEG Stream.
NotEntitledException	This Exception is thrown when the user is not entitled to access the content of the object (the Elementary Stream is scrambled and the user is not entitled).
NothingToAbortException	A NothingToAbortException is thrown when the abort method is called and there is no loading in progress.
NotLoadedException	A NotLoadedException is thrown when an operation fails because information which is required to be loaded has not been.
ServerDeliveryException	A ServerDeliveryException is thrown when the local machine can not communicate with the server.
ServiceXFRException	A ServiceXFRException is thrown when a DSMCC Object can not be loaded in the current ServiceDomain but is available in an alternate ServiceDomain (i.e. for an object Carousel, the IOR of the object or one of its parent directories contains a Lite Option Profile Body).
UnknownEventException	The `UnknownEventException` is thrown when a method tries to access to an unknown event.

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

org.dvb.dsmcc

AsynchronousLoadingEvent

Declaration

```
public abstract class AsynchronousLoadingEvent extends java.util.EventObject

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Direct Known Subclasses:

InsufficientResourcesEvent, InvalidFormatEvent, InvalidPathnameEvent, LoadingAbortedEvent, MPEGDeliveryErrorEvent, NotEntitledEvent, ServerDeliveryErrorEvent, ServiceXFRErrorEvent, SuccessEvent

Description

This class described an Object event which is used to notify the loading of a DSMCC object.

Constructors

AsynchronousLoadingEvent(DSMCCObject)

```
public AsynchronousLoadingEvent(org.dvb.dsmcc.DSMCCObject o)
```

Creates an AsynchronousLoadingEvent.

Parameters:

o - the DSMCCObject that generated the event.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that generated the event.

Overrides:

```
getSource in class EventObject
```

Returns:

the DSMCCObject that generated the event.

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

org.dvb.dsmcc AsynchronousLoadingEventListener

Declaration

```
public interface AsynchronousLoadingEventListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

Listener for applications which perform asynchronous loading, in order to be informed if the loading is done or if an error has occurred.

Methods

receiveEvent(AsynchronousLoadingEvent)

```
public void receiveEvent(org.dvb.dsmcc.AsynchronousLoadingEvent e)
```

Method called when an event is sent to the application.

Parameters:

`e` - an AsynchronousLoadingEvent event.

org.dvb.dsmcc DSMCCException

```
Declaration
public class DSMCCException extends java.io.IOException

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Direct Known Subclasses:

IllegalObjectTypeException, InsufficientResourcesException, InvalidAddressException, InvalidFormatException, InvalidPathNameException, MPEGDeliveryException, NotEntitledException, NothingToAbortException, NotLoadedException, ServerDeliveryException, ServiceXFRException, UnknownEventException

Description

The DSMCCException is the root class of all DSMCC related exceptions.

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

Constructors

DSMCCException()

```
public DSMCCException()
```

Construct a DSMCCException with no detail message.

DSMCCException(String)

```
public DSMCCException(java.lang.String s)
```

Construct a DSMCCException with the specified detail message.

Parameters:

`s` - the detail message.

org.dvb.dsmcc

DSMCCObject

Declaration

```
public class DSMCCObject extends java.io.File
```

```
java.lang.Object
  |
  +--java.io.File
        |
        +--org.dvb.dsmcc.DSMCCObject
```

All Implemented Interfaces:

```
java.lang.Comparable, java.io.Serializable
```

Description

A `DSMCCObject` extends the java.io.File class with a combination of features appropriate for any file system with a long access time and features specific to the DSMCC object carousel as one example of such a file system. Files referenced by a `DSMCCObject` exist within a file system. This may be represented by an instance of `ServiceDomain` but need not be, particularly for the file system in which the initial file of the application itself is carried.

A DSMCCObject is specified by a pathname, which can either be an absolute pathname or a relative pathname. Relative paths shall work as defined in "Broadcast Transport Protocol Access API" in the main body of the present document. Path names must follow the naming conventions of the host platform. The constructors of this class shall accept the absolute paths returned by java.io.File.getAbsolutePath().

To access the content of the object:

- For a Directory, the method list of the java.io.File class has to be used to get the entries of the directory.
- For a Stream object, the class DSMCCStream has to be used.
- For a File, the java.io.FileInputStream class or the java.io.RandomAccessFile has to be used.

NOTE: Obviously, for the Object Carousel, the write mode of java.io.RandomAccessFile is not allowed.

DSMCCObjects exist in two states, loaded and unloaded as returned by the isLoaded method. Transitions from unloaded to loaded are triggered by applications calling the asynchronousLoad or synchronousLoad or getSigners(boolean) methods. Transitions from loaded to unloaded are triggered by applications calling the unload method. Attempting to load an already loaded object does not cause it to be re-loaded.

The only state transitions for a DSMCCObject shall be only in response to these method calls. There shall be no implicit state transitions in either direction. When the application no longer has any references to an object in the loaded state, the system resources allocated should be freed by the system.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 486 of 861

The state machine of DSMCCObject is disconnected from any state model of any cache for the file system that contains it. Objects may appear in that cache without any corresponding DSMCCObject being in the loaded state. Objects which are in that cache and where any corresponding DSMCCObject is not in the loaded state may disappear from that cache at any time. The contents of a object may be accessible to applications from the cache without the DSMCCObject ever being in the loaded state.

NOTE: DSMCCObjects in the loaded state will consume memory in the MHP receiver. If memory in the MHP receiver is short, this memory can only be recovered by the receiver killing the MHP application. Applications which can accept weaker guarantees about the data of a DSMCCObject being available should use the prefetch methods.

See Also:

`ServiceDomain`

Fields

FORCED_STATIC_CACHING

```
public static final int FORCED_STATIC_CACHING
```

Constant to indicate that the data for an object shall be returned from the cache if present, disregarding the cache priority signalling. If the data is not in the cache, it shall be retrieved from the network.

This strategy is intended solely for situations where generic DVB object carousels (as opposed to MHP object carousels) are being accessed and only the accessing application knows which files are static and which are dynamic. It should not be used to access files in MHP object carousels where the Caching priority descriptor can and should be used.

Since:

MHP 1.1.2.

FROM_CACHE

```
public static final int FROM_CACHE
```

Constant to indicate that the data for an object shall only be retrieved where it is already in cache and meets the requirements of cache priority signaling. Where data is not in the cache, or the contents do not meet the requirements of the of cache priority signaling (i.e. cache priority signalling indicates that an object re-acquisition is required), attempts to load a DSMCCObject shall fail with `MPEGDeliveryException` or `MPEGDeliveryErrorEvent` for synchronousLoad and asynchronousLoad respectively.

Since:

MHP 1.0.1.

FROM_CACHE_OR_STREAM

```
public static final int FROM_CACHE_OR_STREAM
```

Constant to indicate that the data for an object shall be automatically be retrieved from the network where the data is not already cached. Note that this method does not modify the caching policy controlled by the signaling in the OC. So, if the data is signalled as requiring transparent caching then data will be retrieved from the network if required.

Since:

MHP 1.0.1.

FROM_STREAM_ONLY

```
public static final int FROM_STREAM_ONLY
```

Constant to indicate that the data for an object shall always be retrieved from the network even if the data has already been cached.

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

NOTE: This functionality is introduced deliberately to bypass any receiver cache allowing an application to be synchronised with the broadcast carousel. It should not be used simply to ensure up-to-date content is retrieved. It should also be noted here that this feature is provided to fulfil very specific application requirements (where other provided methods may not be suitable) and misuse of this functionality may seriously affect application/receiver performance.

Since:

MHP 1.0.1.

Constructors

DSMCCObject(DSMCCObject, String)

```
public DSMCCObject(org.dvb.dsmcc.DSMCCObject dir, java.lang.String name)
```

Create a DSMCCObject object.

Parameters:

`dir` - the directory object.

`name` - the file pathname.

DSMCCObject(String)

```
public DSMCCObject(java.lang.String path)
```

Create a DSMCCObject object.

Parameters:

`path` - the path to the file.

DSMCCObject(String, String)

```
public DSMCCObject(java.lang.String path, java.lang.String name)
```

Create a DSMCCObject object.

Parameters:

`path` - the directory Path.

`name` - the file pathname.

Methods

abort()

```
public void abort()
throws NothingToAbortException
```

This method is used to abort a load in progress. It can be used to abort either a synchronousLoad or an asynchronous-Load.

Throws:

`NothingToAbortException` - There is no loading in progress.

addObjectChangeEventListener(ObjectChangeEventListener)

```
public void addObjectChangeEventListener(org.dvb.dsmcc.ObjectChangeEventListener listener)
throws InsufficientResourcesException
```

Subscribes an ObjectChangeEventListener to receive notifications of version changes of DSMCCObject.

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

This listener shall never be fired until after the object has successfully entered the loaded state for the first time. Hence objects which never successfully enter the loaded state (e.g. because the object cannot be found) shall never have this listener fire. Once an object has successfully entered the loaded state once, this event shall continue to be fired when changes are detected by the MHP regardless of further transitions in or out of the loaded state.

NOTE: The algorithm used for this change monitoring is implementation dependent. In some implementations, this exception will always be thrown. In other implementations, it will never be thrown. In other implementations, whether it is thrown or not will depend on the complexity and design of the object carousel in which the object is carried. Even where no exception is thrown, implementations are not required to detect all possible forms in which an object may change.

Parameters:

`listener` - the ObjectChangeEventListener to be notified.

Throws:

`InsufficientResourcesException` - if there are not sufficient resources to monitor the object for changes.

asynchronousLoad(AsynchronousLoadingEventListener)

```
public void asynchronousLoad(org.dvb.dsmcc.AsynchronousLoadingEventListener l)
throws InvalidPathNameException
```

This method starts the asynchronous loading of the data for a DSMCCObject. This method can fail either asynchronously with an event or synchronously with an exception. When it fails synchronously with an exception, no event shall be sent to the listener. For each call to this method which returns without throwing an exception, one of the following events will be sent to the application (by a listener mechanism) as soon as the loading is done or if an error has occurred: SuccessEvent, InvalidFormatEvent, InvalidPathNameEvent, MPEGDeliveryErrorEvent, ServerDelivery-ErrorEvent, ServiceXFRErrorEvent, NotEntitledEvent, LoadingAbortedEvent, InsufficientResourcesEvent.

Parameters:

`l` - an AsynchronousLoadingEventListener to receive events related to asynchronous loading.

Throws:

`InvalidPathNameException` - the Object can not be found, or the service domain is not in an attached state.

getSigners()

```
public java.security.cert.X509Certificate[][] getSigners()
```

This method shall attempt to validate all certificate chains found for this file in the network. Valid chains do not need to originate from root certificates known to the MHP terminal, e.g. self signing of data files. Applications should note that calls to this method may take some time. If the `DSMCCObject` is not loaded, this method will return null. If the `DSMCCObject` is loaded, this method returns the same as getSigners(false), except if getSigners(false) would throw an exception, this method will return an outer array of size zero.

NOTE: If the file in the network changes between when it was loaded and when the hash file(s), signature & certificate files are read and those files have been updated to match the new version of the file then the hash value of the data which was loaded will not match the hash value in the hash file in the network and hence no certificate chains will be valid.

Returns:

a two-dimensional array of X.509 certificates, where the first index of the array determines a certificate chain and the second index identifies the certificate within the chain. Within one certificate chain the leaf certificate is first followed by any intermediate certificate authorities in the order of the chain with the root CA certificate as the last item.

Since:

MHP 1.0.1.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 489 of 861

getSigners(boolean)

```
public java.security.cert.X509Certificate[][] getSigners(boolean known_root)
throws InvalidFormatException, InterruptedIOException, MPEGDeliveryException, ServerDeliveryExceptio
n, InvalidPathNameException, NotEntitledException, ServiceXFRException, InsufficientResourcesExcepti
on
```

This method shall attempt to validate all certificate chains found for this file in the network. The process of determining validity is the same as the process of authentication except that when known_root is false, checking whether the root certificate is known is not included. The known_root parameter to the method defines whether the MHP terminal shall check if the root certificate in each chain is known to it or not. If the root certificate is checked then chains with unknown root certificates shall not be considered to be valid. If root certificates are not checked then the MHP application is responsible for comparing them with some certificate which it provides (e.g. for self signing of data files). The hash file(s), signature & certificate files shall be shall be fetched from the network in compliance with the caching priority defined in the main body of the present document. If the object is in the loaded state then the data of the file which was loaded shall be used and no new file contents loaded. If the object is not in the loaded state then this method shall attempt to load it as if synchronousLoad had been called. Applications should note that calls to this method may take some time.

NOTE: If the file in the network changes between when it was loaded and when the hash file(s), signature and certificate files are read and those files have been updated to match the new version of the file then the hash value of the data which was loaded will not match the hash value in the hash file in the network and hence no certificate chains will be valid.

Parameters:

`known_root` - if true then valid certificate chains are only those where the root is known to the MHP terminal. If false, the validity of the chain shall be determined without considering whether the root is known to the MHP terminal or not.

Returns:

a two-dimensional array of X.509 certificates, where the first index of the array determines a certificate chain and the second index identifies the certificate within the chain. Within one certificate chain the leaf certificate is first followed by any intermediate certificate authorities in the order of the chain with the root CA certificate as the last item. If no certificate chains are found to be valid then an outer array of size zero shall be returned.

Throws:

`java.io.InterruptedIOException` - the loading has been aborted.

`InvalidPathNameException` - the Object can not be found, or the service domain is not in an attached state.

`NotEntitledException` - the stream carrying the object is scrambled and the user has no entitlements to descramble the stream.

`ServiceXFRException` - the file system containing the DSMCCObject is a DSMCC object carousel and the IOR of the object or one of its parent directories is a Lite Option Profile Body and the referenced carousel is not already mounted by the MHP terminal.

`InvalidFormatException` - an error occured in the format of the file system containing the DSMCCObject, for instance an inconsistent DSMCC message has been received.

`MPEGDeliveryException` - an error has occurred while loading data for the file, for instance a timeout when reading data from an MPEG stream

`ServerDeliveryException` - when an MHP terminal cannot communicate with the server for files delivered over a bi-directional IP connection.

`InsufficientResourcesException` - there is not enough memory to load the object

Since:

MHP 1.0.3.

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

getURL()

```
public java.net.URL getURL()
```

Returns a URL identifying this carousel object. If the directory entry for the object has not been loaded then null shall be returned.

Returns:

a URL identifying the carousel object or null.

Since:

MHP 1.0.1.

isLoaded()

```
public boolean isLoaded()
```

Returns a boolean indicating whether or not the DSMCCObject has been loaded.

Returns:

true if the file is already loaded, false otherwise.

isObjectKindKnown()

```
public boolean isObjectKindKnown()
```

Returns a boolean indicating if the kind of the object is known. (The kind of an object is known if the directory containing it is loaded).

Returns:

true if the type of the object is known, false otherwise.

isStream()

```
public boolean isStream()
```

Returns a boolean indicating whether or not the DSMCCObject is a DSMCC Stream object.

Returns:

true if the file is a stream, false if the object is not a stream or if the object kind is unknown.

isStreamEvent()

```
public boolean isStreamEvent()
```

Returns a boolean indicating whether or not the DSMCCObject is a DSMCC StreamEvent object.

NOTE: If isStreamEvent is true then isStream is true also.

Returns:

true if the file is a stream event, false if the object is not a stream event or if the object kind is unknown.

loadDirectoryEntry(AsynchronousLoadingEventListener)

```
public void loadDirectoryEntry(org.dvb.dsmcc.AsynchronousLoadingEventListener l)
throws InvalidPathNameException
```

Asynchronous loading of the directory entry information. Calling this is equivalent of calling the method `asynchronousLoad` on the parent directory of a `DSMCCObject`. This method can fail either asynchronously with an event or synchronously with an exception. When it fails synchronously with an exception, no event shall be sent to the listener.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 491 of 861

Parameters:

`l` - a listener which will be called when the loading is done.

Throws:

`InvalidPathNameException` - if the object cannot be found.

prefetch(DSMCCObject, String, byte)

```
public static boolean prefetch(org.dvb.dsmcc.DSMCCObject dir, java.lang.String path, byte priority)
```

Calling this method will issue a hint to the MHP for pre-fetching the object data for that DSMCC object into cache.

Parameters:

`dir` - the directory object in which to pre-fetch the data.

`path` - the relative path name of object to pre-fetch, starting from the directory object passes as parameter.

`priority` - the relative priority of this pre-fetch request (higher = more important)

Returns:

true if the MHP supports pre-fetching (i.e. will try to process the request) and false otherwise. Note that a return value of 'true' is only an indication that the MHP receiver supports pre-fetching. It is not a guarantee that the requested data will actually be loaded into cache as the receiver may decide to drop the request in order to make resources available for regular load requests.

prefetch(String, byte)

```
public static boolean prefetch(java.lang.String path, byte priority)
```

Calling this method will issue a hint to the MHP for pre-fetching the object data for that DSMCC object into cache.

Parameters:

`path` - the absolute pathname of the object to pre-fetch.

`priority` - the relative priority of this pre-fetch request (higher = more important)

Returns:

true if the MHP supports pre-fetching (i.e. will try to process the request) and false otherwise. Note that a return value of 'true' is only an indication that the MHP receiver supports pre-fetching. It is not a guarantee that the requested data will actually be loaded into cache as the receiver may decide to drop the request in order to make resources available for regular load requests.

removeObjectChangeEventListener(ObjectChangeEventListener)

```
public void removeObjectChangeEventListener(org.dvb.dsmcc.ObjectChangeEventListener listener)
```

Unsubscribes an ObjectChangeEventListener to receive notifications of version changes of DSMCCObject.

Parameters:

`listener` - a previously registered ObjectChangeEventListener.

setRetrievalMode(int)

```
public void setRetrievalMode(int retrieval_mode)
```

Set the retrieval mode for a `DSMCCObject`. The default retrieval mode is FROM_CACHE_OR_STREAM. The retrieval mode state is sampled when the object is loaded (whether explicitly or as described in "Constraints on the java.io.File methods for broadcast carousels"). Changing the retrieval mode for a loaded object has no effect until the object is unloaded and loaded again.

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

Parameters:

retrieval_mode - the retrieval mode to be used for the object specified as one of the public static final constants in this class.

Throws:

java.lang.IllegalArgumentException - if the retrieval_mode specified is not one listed defined for use with this method.

Since:

MHP 1.0.1.

synchronousLoad()

```
public void synchronousLoad()
throws InvalidFormatException, InterruptedIOException, MPEGDeliveryException, ServerDeliveryExceptio
n, InvalidPathNameException, NotEntitledException, ServiceXFRException, InsufficientResourcesExcepti
on
```

This method is used to load a DSMCCObject. This method blocks until the file is loaded. It can be aborted from another thread with the abort method. In this case the InterruptedIOException is thrown. For DSMCCObjects contained within a DSMCC object carousel, if the IOR of the object itself or one of its parent directories is a Lite Option Profile Body and the referenced carousel is not attached, the MHP implementation will not attempt to resolve it: a ServiceXFRException is thrown to indicate to the application where the DSMCCObject is actually located.

Throws:

java.io.InterruptedIOException - the loading has been aborted.

InvalidPathNameException - the Object can not be found, or the service domain is not in an attached state.

NotEntitledException - the stream carrying the object is scrambled and the user has no entitlements to descramble the stream.

ServiceXFRException - the file system containing the DSMCCObject is a DSMCC object carousel and the IOR of the object or one of its parent directories is a Lite Option Profile Body and the referenced carousel is not already mounted by the MHP terminal.

InvalidFormatException - an error occured in the format of the file system containing the DSMCCObject, for instance an inconsistent DSMCC message has been received.

MPEGDeliveryException - an error has occurred while loading data for the file, for instance a timeout when reading data from an MPEG stream

ServerDeliveryException - when an MHP terminal cannot communicate with the server for files delivered over a bi-directional IP connection.

InsufficientResourcesException - there is not enough memory to load the object

unload()

```
public void unload()
throws NotLoadedException
```

When calling this method, the applications gives a hint to the MHP that if this object is not consumed by another application/thread, the system can free all the resources allocated to this object. It is worth noting that if other clients use this object (e.g. a file input stream is opened on this object or if the corresponding stream or stream event is being consumed) the system resources allocated to this object will not be freed. This method puts the DSMCCObject into the unloaded state.

Throws:

NotLoadedException - the carousel object is not loaded.

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

org.dvb.dsmcc

DSMCCStream

Declaration

```
public class DSMCCStream

java.lang.Object
  |
  +--org.dvb.dsmcc.DSMCCStream
```

Direct Known Subclasses:

DSMCCStreamEvent

Description

The DSMCCStream class is used to manage DSMCC Stream Objects. The BIOP::Stream message shall be read from the network once only, before the constructor of this class returns. Hence methods which return information from that message shall not be affected by any subsequent changes to that information.

The methods in this class relating to the NPT mechanism shall also be mapped onto the DVB timeline as defined in the main part of the present document.

NOTE: The NPT mechanism and scheduled stream events that depend on it are known to be vulnerable to disruption in many digital TV distribution networks. Existing deployed network equipment that re-generates the STC is unlikely to be aware of NPT and hence will not make the necessary corresponding modification to STC values inside NPT reference descriptors. Applications should only use NPT where they are confident that the network where they are to be used does not have this problem.

See Also:

DSMCCObject

Constructors

DSMCCStream(DSMCCObject)

```
public DSMCCStream(org.dvb.dsmcc.DSMCCObject aDSMCCObject)
throws NotLoadedException, IllegalObjectTypeException
```

Creates a Stream Object from a DSMCC Object. The BIOP message referenced by the DSMCCObject has to be a Stream or StreamEvent BIOP message.

Parameters:

aDSMCCObject - the DSMCC object which describes the stream

Throws:

NotLoadedException - the DSMCCObject is not loaded.

IllegalObjectTypeException - the DSMCCObject is neither a DSMCC Stream nor a DSMCCStream-Event.

DSMCCStream(String)

```
public DSMCCStream(java.lang.String path)
throws IOException, IllegalObjectTypeException
```

Create a Stream Object from its pathname. For an object Carousel, this method will block until the module which contains the object is loaded. The BIOP message referenced by the DSMCCObject pointed to by the parameter path has to be a Stream or StreamEvent BIOP message.

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

Parameters:

path - the pathname of the DSMCCStream Object.

Throws:

java.io.IOException - If an IO error occurred.

IllegalObjectTypeException - the DSMCCObject is neither a DSMCC Stream nor a DSMCCStream-Event.

DSMCCStream(String, String)

```
public DSMCCStream(java.lang.String path, java.lang.String name)
throws IOException, IllegalObjectTypeException
```

Create a DSMCCStream from its pathname. For an object Carousel, this method will block until the module which contains the object is loaded. The BIOP message referenced by the DSMCCObject pointed to be the parameters path and name has to be a Stream or StreamEvent BIOP message.

Parameters:

path - the directory path.

name - the name of the DSMCCStream Object.

Throws:

java.io.IOException - If an IO error occurred.

IllegalObjectTypeException - the DSMCCObject is neither a DSMCC Stream nor a DSMCCStream-Event.

Methods

addNPTListener(NPTListener)

```
public void addNPTListener(org.dvb.dsmcc.NPTListener l)
```

Add a listener for timeline events on the DSMCCStream object. Adding the same listener a second time has no effect.

Parameters:

l - the listener.

Since:

MHP 1.0.1.

getDuration()

```
public long getDuration()
```

This function returns the duration in milliseconds of the DSMCC Stream. If the DSMCCStream BIOP message does not specify duration, zero will be returned.

Returns:

The duration in milliseconds of the DSMCC Stream.

getMediaTimeFromStream(long, Clock)

```
public javax.media.Time getMediaTimeFromStream(long time, javax.media.Clock c)
```

Convert a time measured according to a timeline in the stream into a JMF media time measured in a specified clock.

Parameters:

time - the time measured according to the timeline in the stream.

c - the clock to express the media time.

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

Returns:

the JMF media time corresponding to the specified time.

Throws:

`java.lang.IllegalArgumentException` - if the Clock is not a Player which is presenting at least one of the streams of this collection (as returned by getStreamLocator).

getNPT()

```
public long getNPT()
throws MPEGDeliveryException
```

This function is used to get value of the current timeline in milliseconds. Implementations are not required to continuously monitor for the timeline descriptors. In implementations which do not continuously monitor, this method will block while the current timeline is retrieved from the stream.

Returns:

the current time in milliseconds or zero if the DSMCC Stream object BIOP message does not contain any taps pointing to a timeline.

Throws:

`MPEGDeliveryException` - if there is an error in retrieving the timeline descriptors.

getNPTRate()

```
public org.dvb.dsmcc.NPTRate getNPTRate()
throws MPEGDeliveryException
```

Get the rate for the timeline referenced in the DSMCCStream object. Where the timeline is NPT, the NPT rate is returned. Where the timeline is the DVB timeline, 0/1 is returned if it is paused and 1/1 is returned if it is advancing. The method returns null if no timeline is referenced by the DSMCCStream object, (i.e. for NPT, no STR_NPT_USE tap in the list of taps).

Returns:

the NPT rate or null.

Throws:

throws - MPEGDeliveryException if there is an error in retrieving NPT reference descriptors.

`MPEGDeliveryException`

Since:

MHP 1.0.1.

getStreamLocator()

```
public org.davic.net.Locator getStreamLocator()
```

This function returned a Locator referencing the streams of this collection. The interpretation of the return value is determined by the `isMPEGProgram` method.

Returns:

a locator.

isAudio()

```
public boolean isAudio()
```

This function returns a boolean indicating if the Stream Object refers to an audio stream. This is the case if the audio field in the Stream(Event) BIOP message has a value different from zero.

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

Returns:

true only if the Stream object refers to an audio stream.

isData()

```
public boolean isData()
```

This function returns a boolean indicating if the Stream Object refers to a data stream. This is the case if the data field in the Stream(Event) BIOP message has a value different from zero.

Returns:

true only if the Stream object refers to a data stream.

isMPEGProgram()

```
public boolean isMPEGProgram()
```

This method will return true if the Stream(Event) BIOP message contains a tap with use field BIOP_PROGRAM_USE, otherwise it will return false.

Returns:

true only if the Stream(Event) BIOP message is as described above.

isVideo()

```
public boolean isVideo()
```

This function returns a boolean indicating if the Stream Object refers to an video stream. This is the case if the `video' field in the Stream(Event) BIOP message has a value different from zero otherwise false is returned.

Returns:

true only if the Stream object refers to an video stream.

removeNPTListener(NPTListener)

```
public void removeNPTListener(org.dvb.dsmcc.NPTListener l)
```

Remove a listener for timeline events on the `DSMCCStream` object. Removing a non-subscribed listener has no effect.

Parameters:

`l` - the listener to remove.

Since:

MHP 1.0.1.

org.dvb.dsmcc

DSMCCStreamEvent

Declaration

```
public class DSMCCStreamEvent extends DSMCCStream
```

```
java.lang.Object
  |
  +--org.dvb.dsmcc.DSMCCStream
        |
        +--org.dvb.dsmcc.DSMCCStreamEvent
```


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

Description

The DSMCCStreamEvent class is used to manage DSMCC StreamEvent Objects. Applications wishing to monitor changes in the list of events which are part of this stream event should use `DSMCCObject.addObjectChangeEvent-Listener` on the `DSMCCObject` representing which describes this set of stream events. The BIOP::StreamEvent message shall be read from the network once only, before the constructor of this class returns. Hence methods which return information from that message shall not be affected by any subsequent changes to that information.

The subscribe method only verifies that the event name can be bound to an eventId but it does not require that the stream event descriptors for that event id can be received at that moment. While the event listener is registered the MHP terminal shall filter the stream event descriptors as specified in the "Monitoring broadcast timebases and events" clause in the main body of the specification.

NOTE: The NPT mechanism and scheduled stream events that depend on it are known to be vulnerable to disruption in many digital TV distribution networks. Existing deployed network equipment that re-generates the STC is unlikely to be aware of NPT and hence will not make the necessary corresponding modification to STC values inside NPT reference descriptors. Applications should only use NPT where they are confident that the network where they are to be used does not have this problem.

Constructors

DSMCCStreamEvent(DSMCCObject)

```
public DSMCCStreamEvent(org.dvb.dsmcc.DSMCCObject aDSMCCObject)
throws NotLoadedException, IllegalObjectTypeException
```

Create a `DSMCCStreamEvent` from a `DSMCCObject`. The Object has to be a DSMCC StreamEvent.

Parameters:

`aDSMCCObject` - the DSMCC object which describes the stream.

Throws:

`NotLoadedException` - the DSMCCObject is not loaded.

`IllegalObjectTypeException` - the DSMCCObject does not lead to a DSMCC StreamEvent.

DSMCCStreamEvent(String)

```
public DSMCCStreamEvent(java.lang.String path)
throws IOException, IllegalObjectTypeException
```

Create a DSMCCStreamEvent Object from its pathname. The path has to lead to a DSMCCStreamEvent.

Parameters:

`path` - the pathname of the DSMCCStreamEvent object.

Throws:

`java.io.IOException` - An IO error has occurred.

`IllegalObjectTypeException` - the path does not lead to a DSMCC StreamEvent.

DSMCCStreamEvent(String, String)

```
public DSMCCStreamEvent(java.lang.String path, java.lang.String name)
throws IOException, IllegalObjectTypeException
```

Create a DSMCCStreamEvent from its pathname. For an object Carousel, this method will block until the module which contains the object is loaded. The path has to lead to a DSMCC Stream Event.

Parameters:

`path` - the directory path.

`name` - the name of the DSMCCStreamEvent Object.

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

Throws:

java.io.IOException - If an IO error occurred.

IllegalObjectTypeException - the path does not lead to a DSMCC StreamEvent.

Methods

getEventList()

```
public java.lang.String[] getEventList()
```

This function is used to get the list of the events of the DSMCCStreamEvent object.

Returns:

The list of the eventName.

subscribe(String, StreamEventListener)

```
public int subscribe(java.lang.String eventName, org.dvb.dsmcc.StreamEventListener l)
throws UnknownEventException, InsufficientResourcesException
```

This function is used to subscribe to an event of a DSMCC StreamEvent object.

Parameters:

eventName - the name of the event.

l - an object that implements the StreamEventListener Interface.

Returns:

The event Identifier.

Throws:

UnknownEventException - the event cannot be found at this time.

InsufficientResourcesException - if resources needed to perform the subscription are not available.

unsubscribe(int, StreamEventListener)

```
public void unsubscribe(int eventId, org.dvb.dsmcc.StreamEventListener l)
throws UnknownEventException
```

This function is used to cancel the subscription to an event of a DSMCCEvent object.

Parameters:

eventId - Identifier of the event.

l - an object that implements the StreamEventListener Interface.

Throws:

UnknownEventException - The event can not be found.

unsubscribe(String, StreamEventListener)

```
public void unsubscribe(java.lang.String eventName, org.dvb.dsmcc.StreamEventListener l)
throws UnknownEventException
```

This function is used to cancel the subscription to an event of a DSMCCEvent object.

Parameters:

eventName - the name of the event.

l - an object that implements the StreamEventListener Interface.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 499 of 861

Throws:

UnknownEventException - The event can not be found.

org.dvb.dsmcc IllegalObjectTypeException

Declaration

```
public class IllegalObjectTypeException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.IllegalObjectTypeException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This Exception is thrown when the application attempted to create a DSMCCStream or DSMCCStreamEvent object with an object or a path that did not correspond to a DSMCC Stream or DSMCC StreamEvent respectively.

Constructors

IllegalObjectTypeException()

```
public IllegalObjectTypeException()
```

Constructor of the exception with no detail message.

IllegalObjectTypeException(String)

```
public IllegalObjectTypeException(java.lang.String s)
```

Constructor of the exception.

Parameters:

s - detail message.

org.dvb.dsmcc InsufficientResourcesEvent

Declaration

```
public class InsufficientResourcesEvent extends AsynchronousLoadingEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
              |
              +--org.dvb.dsmcc.InsufficientResourcesEvent
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 500 of 861

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This event is generated if there are insufficient resources available to load a DSMCCObject. e.g. if there is not enough memory.

Constructors

InsufficientResourcesEvent(DSMCCObject)

```
public InsufficientResourcesEvent(org.dvb.dsmcc.DSMCCObject o)
```

Create an InsufficientResourcesException object.

Parameters:

- o - the DSMCCObject that generated the event.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that generated the event.

Overrides:

getSource in class AsynchronousLoadingEvent

Returns:

the DSMCCObject that generated the event.

org.dvb.dsmcc

InsufficientResourcesException

Declaration

```
public class InsufficientResourcesException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.InsufficientResourcesException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This exception gets thrown when a request to perform a DSMCC related operation cannot be completed due to resource limitations. For example, no section filters or system timers may be available.

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

Since:

MHP 1.0.1.

Constructors

InsufficientResourcesException()

```
public InsufficientResourcesException()
```

Construct an InsufficientResourcesException with no detail message.

InsufficientResourcesException(String)

```
public InsufficientResourcesException(java.lang.String message)
```

Construct an InsufficientResourcesException with the specified detail message.

Parameters:

`message` - the message for the exception.

org.dvb.dsmcc

InvalidAddressException

Declaration

```
public class InvalidAddressException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.InvalidAddressException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

An InvalidAddressException is thrown when the format of an NSAP address is not recognized.

Constructors

InvalidAddressException()

```
public InvalidAddressException()
```

Construct a InvalidAddressException with no detail message.

InvalidAddressException(String)

```
public InvalidAddressException(java.lang.String s)
```

Construct a InvalidAddressException with the specified detail message.

Parameters:

`s` - the detail message.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 502 of 861

org.dvb.dsmcc

InvalidFormatEvent

Declaration

```
public class InvalidFormatEvent extends AsynchronousLoadingEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
              |
              +--org.dvb.dsmcc.InvalidFormatEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This event is generated if the format of the data received is inconsistent.

Constructors

InvalidFormatEvent(DSMCCObject)

```
public InvalidFormatEvent(org.dvb.dsmcc.DSMCCObject o)
```

Create an InvalidFormatException object.

Parameters:

o - the DSMCCObject that generated the event.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that generated the event.

Overrides:

getSource in class AsynchronousLoadingEvent

Returns:

the DSMCCObject that generated the event.

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

org.dvb.dsmcc InvalidFormatException

Declaration

```
public class InvalidFormatException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.InvalidFormatException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

An InvalidFormatException is thrown when an inconsistent DSMCC message is received.

Constructors

InvalidFormatException()

```
public InvalidFormatException()
```

Construct an InvalidFormatException with no detail message.

InvalidFormatException(String)

```
public InvalidFormatException(java.lang.String s)
```

Construct an InvalidFormatException with the specified detail message.

Parameters:

`s` - the detail message.

org.dvb.dsmcc InvalidPathnameEvent

Declaration

```
public class InvalidPathnameEvent extends AsynchronousLoadingEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
              |
              +--org.dvb.dsmcc.InvalidPathnameEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```


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

Description

The pathname does not exist or the ServiceDomain has been detached.

Constructors

InvalidPathnameEvent(DSMCCObject)

```
public InvalidPathnameEvent(org.dvb.dsmcc.DSMCCObject o)
```

Create an InvalidPathnameEvent.

Parameters:

- o - the DSCMCCObject that generated this event.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that generated the event.

Overrides:

getSource in class AsynchronousLoadingEvent

Returns:

the DSMCCObject that generated the event.

org.dvb.dsmcc

InvalidPathNameException

Declaration

```
public class InvalidPathNameException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.InvalidPathNameException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

The InvalidPathNameException is thrown when the path name to a DSMCCObject does not exist or if the service domain is not detached.

Constructors

InvalidPathNameException()

```
public InvalidPathNameException()
```

Construct an InvalidPathNameException with no detail message.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 505 of 861

InvalidPathNameException(String)

```
public InvalidPathNameException(java.lang.String s)
```

Construct an InvalidPathNameException with the specified detail message.

Parameters:

s - the detail message.

org.dvb.dsmcc

LoadingAbortedEvent

Declaration

```
public class LoadingAbortedEvent extends AsynchronousLoadingEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
              |
              +--org.dvb.dsmcc.LoadingAbortedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This event will be sent to the AsynchronousEventListener when an asynchronous loading operation is aborted.

Since:

MHP 1.0.1.

Constructors

LoadingAbortedEvent(DSMCCObject)

```
public LoadingAbortedEvent(org.dvb.dsmcc.DSMCCObject aDSMCCObject)
```

Creates a LoadingAbortedEvent object.

Parameters:

aDSMCCObject - the DSMCCObject that generated the event.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that generated the event.

Overrides:

getSource in class AsynchronousLoadingEvent.

Returns:

the DSMCCObject whose loading was aborted.

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

org.dvb.dsmcc

MPEGDeliveryErrorEvent

Declaration

```
public class MPEGDeliveryErrorEvent extends AsynchronousLoadingEvent

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
              |
              +--org.dvb.dsmcc.MPEGDeliveryErrorEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

An MPEGDeliveryErrorEvent indicates that an error (for instance, a time out or accessing the data would require tuning) has occurred while loading data from an MPEG Stream.

Constructors

MPEGDeliveryErrorEvent(DSMCCObject)

```
public MPEGDeliveryErrorEvent(org.dvb.dsmcc.DSMCCObject o)
```

Creates an MPEGDeliveryEvent.

Parameters:

- o - the DSMCCObject that generated the event.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that generated the event.

Overrides:

getSource in class AsynchronousLoadingEvent

Returns:

the DSMCCObject that generated the event.

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

org.dvb.dsmcc
MPEGDeliveryException

Declaration

```
public class MPEGDeliveryException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.MPEGDeliveryException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

An MPEGDEliveryException is thrown when an error (for instance, a time out or accessing the data would require tuning) occurs while loading data from an MPEG Stream.

Constructors

MPEGDeliveryException()

```
public MPEGDeliveryException()
```

Construct an MPEGDeliveryException with no detail message.

MPEGDeliveryException(String)

```
public MPEGDeliveryException(java.lang.String s)
```

Construct an MPEGDeliveryException with the specified detail message.

Parameters:

`s` - the detail message.

org.dvb.dsmcc
NotEntitledEvent

Declaration

```
public class NotEntitledEvent extends AsynchronousLoadingEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
              |
              +--org.dvb.dsmcc.NotEntitledEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```


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

Description

This event is sent when an attempt to asynchronously load an object has failed because the elementary stream carrying the object is scrambled and the user is not entitled to access the content of the object.

Constructors

NotEntitledEvent(DSMCCObject)

```
public NotEntitledEvent(org.dvb.dsmcc.DSMCCObject o)
```

Creates a NotEntitledEvent object.

Parameters:

- o - the DSMCCObject that generated the event.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that generated the event.

Overrides:

getSource in class AsynchronousLoadingEvent

Returns:

the DSMCCObject that generated the event.

org.dvb.dsmcc

NotEntitledException

Declaration

```
public class NotEntitledException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.NotEntitledException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This Exception is thrown when the user is not entitled to access the content of the object (the Elementary Stream is scrambled and the user is not entitled).

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

Constructors

NotEntitledException()

```
public NotEntitledException()
```

Construct a NotEntitledException with no detail message.

NotEntitledException(String)

```
public NotEntitledException(java.lang.String s)
```

Construct a NotEntitledException with a detail message.

Parameters:

s - detail message.

org.dvb.dsmcc

NothingToAbortException

Declaration

```
public class NothingToAbortException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.NothingToAbortException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

A NothingToAbortException is thrown when the abort method is called and there is no loading in progress.

Constructors

NothingToAbortException()

```
public NothingToAbortException()
```

Construct a NothingToAbortException with no detail message.

NothingToAbortException(String)

```
public NothingToAbortException(java.lang.String s)
```

Construct a NothingToAbortException with the specified detail message.

Parameters:

s - the detail message.

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

org.dvb.dsmcc NotLoadedException

Declaration

```
public class NotLoadedException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.NotLoadedException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

A NotLoadedException is thrown when an operation fails because information which is required to be loaded has not been.

Constructors

NotLoadedException()

```
public NotLoadedException()
```

Construct a NotLoadedException with no detail message.

NotLoadedException(String)

```
public NotLoadedException(java.lang.String s)
```

Construct a NotLoadedException with the specified detail message.

Parameters:

`s` - the detail message.

org.dvb.dsmcc NPTDiscontinuityEvent

Declaration

```
public class NPTDiscontinuityEvent extends NPTStatusEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.NPTStatusEvent
              |
              +--org.dvb.dsmcc.NPTDiscontinuityEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```


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

Description

Sent when an MHP terminal detects a permanent value discontinuity in a broadcast timebase as defined in the main body of the present document. This may represent an error condition in the incoming broadcast.

Where the timebase is delivered using NPT, this event shall be sent following a PCR discontinuity when the MHP terminal has enough information to determine that there will be an NPT discontinuity. If the `NPTDiscontinuity-Event` is because of invalid data in a new NPTReferenceDescriptor then the event will be generated when that new NPTReferenceDescriptor is detected by the MHP terminal. If the `NPTDiscontinuityEvent` is because no new NPTReferenceDescriptor is detected within the time allowed by the main body of the present document then it will be generated when that time interval has elapsed.

Since:

MHP 1.0.1.

Constructors

NPTDiscontinuityEvent(DSMCCStream, long, long)

```
public NPTDiscontinuityEvent(org.dvb.dsmcc.DSMCCStream source, long before, long after)
```

Construct an event. The `before` and `after` values used shall be the values at the time when the receiver determined that a NPT discontinuity has happened. If the `NPTDiscontinuityEvent` is because of invalid data in a new NPTReference-Descriptor then this is the time when that new descriptor was known to be invalid. If `NPTDiscontinuityEvent` is because of the absence of a new NPTReferenceDescriptor then this will be when the MHP terminal detects that the time interval allowed by the present document for such new descriptors has elapsed. Where an NPT value is unknown or cannot be computed, -1 shall be used.

Parameters:

`source` - the stream whose NPT suffered a discontinuity.

`before` - the last NPT value detected before the discontinuity or -1 for DVB timelines.

`after` - the first NPT value detected after the discontinuity or -1 for DVB timelines.

Methods

getFirstNPT()

```
public long getFirstNPT()
```

Return the first known stable value of NPT after the discontinuity.

Returns:

an NPT value or -1 for DVB timelines.

getLastNPT()

```
public long getLastNPT()
```

Return the last known stable value of NPT before the discontinuity.

Returns:

an NPT value or -1 for DVB timelines.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 512 of 861

org.dvb.dsmcc
NPTListener

Declaration

```
public interface NPTListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

Objects that implement the `NPTListener` interface can receive `NPTStatusEvent` and `NPTRateChangedEvent` events.

Since:

MHP 1.0.1.

Methods

receiveNPTStatusEvent(NPTStatusEvent)

```
public void receiveNPTStatusEvent(org.dvb.dsmcc.NPTStatusEvent e)
```

Send a `NPTStatusEvent` to a registered listener.

Parameters:

`e` - a `NPTStatusEvent` describing the status change.

receiveRateChangedEvent(NPTRateChangeEvent)

```
public void receiveRateChangedEvent(org.dvb.dsmcc.NPTRateChangeEvent e)
```

Send a `NPTRateChangeEvent` to a registered listener.

Parameters:

`e` - the `NPTRateChangeEvent` event.

org.dvb.dsmcc
NPTPresentEvent

Declaration

```
public class NPTPresentEvent extends NPTStatusEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.NPTStatusEvent
              |
              +--org.dvb.dsmcc.NPTPresentEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```


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

Description

Sent to listeners on a DSMCCStream object when a broadcast timeline newly appears for that DSMCC stream when it was not previously present. This is specific to the particular timebase for this stream.

Since:

MHP 1.0.1.

Constructors

NPTPresentEvent(DSMCCStream)

```
public NPTPresentEvent(org.dvb.dsmcc.DSMCCStream source)
```

Construct an event.

Parameters:

`source` - the DSMCCStream for which the broadcast timeline appeared.

org.dvb.dsmcc

NPTRate

Declaration

```
public class NPTRate
```

```
java.lang.Object
  |
  +--org.dvb.dsmcc.NPTRate
```

Description

Represents the rate at which an broadcast timeline progresses. Rates are expressed as the combination of a numerator and a denominator. Instances of this class are constructed by the platform and returned to applications. A rate of 1/1 shall indicate "the standard presentation rate" as defined in the NPT specification.

Since:

MHP 1.0.1.

Methods

getDenominator()

```
public int getDenominator()
```

Get the NPT rate's denominator.

Returns:

the denominator.

getNumerator()

```
public int getNumerator()
```

Get the NPT rate's numerator. A value of zero indicates that the NPT is not progressing.

Returns:

the numerator.

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

org.dvb.dsmcc
NPTRateChangeEvent

Declaration

```
public class NPTRateChangeEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.NPTRateChangeEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Sent only when the rate of an broadcast timeline changes value.

Since:

MHP 1.0.1.

Constructors

NPTRateChangeEvent(DSMCCStream, NPTRate)

```
public NPTRateChangeEvent(org.dvb.dsmcc.DSMCCStream source, org.dvb.dsmcc.NPTRate rate)
```

Construct an event.

Parameters:

`source` - the stream whose rate changed.

`rate` - the new rate of that stream immediately following the change.

Methods

getRate()

```
public org.dvb.dsmcc.NPTRate getRate()
```

Return the new rate of the stream immediately after the change.

Returns:

a `NPTRate` object encapsulating the new rate.

getSource()

```
public java.lang.Object getSource()
```

Return the stream whose rate changed.

Overrides:

`getSource` in class `EventObject`

Returns:

the `DSMCCStream` object on which the rate change has occurred.

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

org.dvb.dsmcc
NPTRemovedEvent

Declaration

```
public class NPTRemovedEvent extends NPTStatusEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.NPTStatusEvent
              |
              +--org.dvb.dsmcc.NPTRemovedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Sent to listeners on a DSMCCStream object when a broadcast timeline which was present for that DSMCC stream is removed. This is specific to the particular timebase for this stream.

Since:

MHP 1.0.1.

Constructors

NPTRemovedEvent(DSMCCStream)

```
public NPTRemovedEvent(org.dvb.dsmcc.DSMCCStream source)
```

Construct an event.

Parameters:

`source` - the DSMCCStream from which the broadcast timeline was removed.

org.dvb.dsmcc
NPTStatusEvent

Declaration

```
public abstract class NPTStatusEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.NPTStatusEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Direct Known Subclasses:

```
NPTDiscontinuityEvent, NPTPresentEvent, NPTRemovedEvent
```

Description

Sent when an MHP terminal detects a change of status in the broadcast timeline of a stream.

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

Since:

MHP 1.0.1.

Constructors

NPTStatusEvent(DSMCCStream)

```
public NPTStatusEvent(org.dvb.dsmcc.DSMCCStream source)
```

Construct an event.

Parameters:

`source` - the stream whose broadcast timeline status changed.

Methods

getSource()

```
public java.lang.Object getSource()
```

Return the stream whose broadcast timeline status changed.

Overrides:

`getSource` in class `EventObject`

Returns:

the `DSMCCStream` whose status changed.

org.dvb.dsmcc

ObjectChangeEvent

Declaration

```
public class ObjectChangeEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.ObjectChangeEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This class describes an object change event that is used to monitor the arrival of a new version of a `DSMCCObject`. For files carried in a DSMCC object carousel, when a change in a module is detected, this event shall be sent to all registered listeners for all objects carried in that module.

Constructors

ObjectChangeEvent(DSMCCObject, int)

```
public ObjectChangeEvent(org.dvb.dsmcc.DSMCCObject source, int aVersionNumber)
```

Creates an ObjectChangeEvent indicating that a new version of the monitored DSMCC Object has been detected. It is up to the application to reload the new version of the object.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 517 of 861

Parameters:

`source` - the DSMCCObject whose version has changed.

`aVersionNumber` - the new version number.

Methods

getNewVersionNumber()

```
public int getNewVersionNumber()
```

This method is used to get the new version number of the monitored `DSMCCObject`. For files carried in a DSMCC object carousel, this method shall return the version number of the module carrying the file.

Returns:

the new version number.

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that has changed.

Overrides:

`getSource` in class `EventObject`

Returns:

the DSMCCObject that has changed.

org.dvb.dsmcc

ObjectChangeEventListener

Declaration

```
public interface ObjectChangeEventListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

The objects that implements the ObjectChangeEventListener interface can receive ObjectChangeEvent event.

Methods

receiveObjectChangeEvent(ObjectChangeEvent)

```
public void receiveObjectChangeEvent(org.dvb.dsmcc.ObjectChangeEvent e)
```

Send a ObjectChangeEvent to the ObjectChangeEventListener.

Parameters:

`e` - the ObjectChangeEvent event.

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

org.dvb.dsmcc

ServerDeliveryErrorEvent

Declaration

```
public class ServerDeliveryErrorEvent extends AsynchronousLoadingEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
              |
              +--org.dvb.dsmcc.ServerDeliveryErrorEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

The local machine can not communicate with the server. This event is only used with files implemented by delivery over bi-directional IP connections. For the object carousel the `MPEGDeliveryErrorEvent` is used instead.

Constructors

ServerDeliveryErrorEvent(DSMCCObject)

```
public ServerDeliveryErrorEvent(org.dvb.dsmcc.DSMCCObject o)
```

Creates a ServerDeliveryEvent object.

Parameters:

`o` - the DSMCCObject that generated the event.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that generated the event.

Overrides:

`getSource` in class `AsynchronousLoadingEvent`

Returns:

the DSMCCObject that generated the event.

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

org.dvb.dsmcc ServerDeliveryException

Declaration

```
public class ServerDeliveryException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.ServerDeliveryException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

A ServerDeliveryException is thrown when the local machine can not communicate with the server. This exception is only used with files implemented by delivery over a bi-directional IP connection. For the object carousel the `MPEGDeliveryException` is used instead.

Constructors

ServerDeliveryException()

```
public ServerDeliveryException()
```

Construct a ServerDeliveryException with no detail message.

ServerDeliveryException(String)

```
public ServerDeliveryException(java.lang.String s)
```

Construct a ServerDeliveryException with the specified detail message.

Parameters:

`s` - the detail message.

org.dvb.dsmcc ServiceDomain

Declaration

```
public class ServiceDomain
```

```
java.lang.Object
  |
  +--org.dvb.dsmcc.ServiceDomain
```

Description

A `ServiceDomain` represents a group of DSMCC objects. The objects are sent either using the object carousel for a broadcast network or with the DSM-CC User-to-User protocol for an interactive network.

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

To access the objects of a ServiceDomain, it has to be attached to the file system name space of the MHP terminal. To access the content of an object, the application has four ways:

- It can instantiate the class that is used to read the object (java.io.FileInputStream or java.io.RandomAccessFile for a File or DSMCCStream for a stream) from its pathname. The loading of the object is implicit but the application has no way to abort it. NB: Obviously, for the Object Carousel, the write mode of java.io.RandomAccessFile is not allowed.
- It can instantiate a DSMCCObject and carry out a Synchronous loading. The loading can be aborted by the abort method of the DSMCCObject class. When the object is loaded, the application will instantiate the class used to read the object.
- It can instantiate a DSMCCObject and carry out an Asynchronous loading. So several loading can be started in parallel from the same thread.
- It is also possible to create directly a java.io.File for a DSMCC object.

Instances of ServiceDomain exist in two states, attached and detached. Newly created instances are always in the detached state. They become attached when a call to the attach method succeeds. They become detached following a call to the detach method.

When service domains in the attached state temporarily lose their network connection (e.g. if the MHP terminal tunes away from the transport stream where they are carried), the behaviour of DSMCC objects which are part of the service domain is specified in the main body of the present document. If such a network connection becomes available again then the service domain shall resume normal behaviour.

A service domain which is temporarily lost its network connection may be forced into the detached state by the implementation if the loss of the network connection becomes irrecoverable. The precise details of when this happens are implementation dependent. This is the only situation when shall be forced into the detached state. Once a ServiceDomain is detached, it will never be automatically attached.

Constructors

ServiceDomain()

```
public ServiceDomain()
```

Creates a ServiceDomain object.

Methods

attach(byte[])

```
public void attach(byte[] NSAPAddress)
throws DSMCCException, InterruptedIOException, InvalidAddressException, MPEGDeliveryException
```

This function is used to attach a ServiceDomain from either an object carousel or from an interactive network. This call will block until the attachment is done.

Calling this method on a ServiceDomain object already in the attached state shall imply a detach of the ServiceDomain object before the attach operation unless the ServiceDomain is already attached to the correct location. Hence if the attach operation fails, the appropriate exception for the failure mode shall be thrown and the ServiceDomain is left in a detached state and not attached to the former object carousel / service domain. If the ServiceDomain is already attached to the correct location then the method call shall have no effect.

Parameters:

NSAPAddress - The NSAP Address of a ServiceDomain as defined in in ISO/IEC 13818-6 [38].

Throws:

java.io.InterruptedIOException - The attachment has been aborted.

InvalidAddressException - The NSAP Address is invalid.

DSMCCException - An error has occurred during the attachment.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 521 of 861

MPEGDeliveryException - attaching to this domain would require tuning.

attach(Locator)

```
public void attach(org.davic.net.Locator l)
throws DSMCCException, InterruptedIOException, MPEGDeliveryException
```

This function is used to attach a ServiceDomain from an object carousel. It loads the module which contains the service gateway object and mounts the ServiceDomain volume in the file system hierarchy. This call will block until the service gateway is loaded. It can be aborted by another thread with the method detach. In this case an InterruptedIOException is thrown.

Calling this method on a ServiceDomain object already in the attached state shall imply a detach of the ServiceDomain object before the attach operation unless the ServiceDomain is already attached to the correct location. Hence if the attach operation fails, the appropriate exception for the failure mode shall be thrown and the ServiceDomain is left in a detached state and not attached to the former object carousel/service domain. If the ServiceDomain is already attached to the correct location then the method call shall have no effect.

Parameters:

l - The locator pointing to the elementary stream carrying the DSI of the object carousel, or to a DVB service that carries one and only one object carousel.

Throws:

DSMCCException - An error has occurred during the attachment. For example, the locator does not point to a component carrying a DSI of an Object Carousel or to a service containing a single carousel.

java.io.InterruptedIOException - The attachment has been aborted.

MPEGDeliveryException - attaching to this domain would require tuning.

attach(Locator, int)

```
public void attach(org.davic.net.Locator aDVBService, int aCarouselId)
throws ServiceXFRException, InterruptedIOException, MPEGDeliveryException
```

This function is used to attach a ServiceDomain from an object carousel. It loads the module which contains the service gateway object and mounts the ServiceDomain volume in the file system hierarchy. This call will block until the service gateway is loaded. It can be aborted by another thread with the method detach. In this case an InterruptedIOException is thrown.

Calling this method on a ServiceDomain object already in the attached state shall imply a detach of the ServiceDomain object before the attach operation unless the ServiceDomain is already attached to the correct location. Hence if the attach operation fails, the appropriate exception for the failure mode shall be thrown and the ServiceDomain is left in a detached state and not attached to the former object carousel/service domain. If the ServiceDomain is already attached to the correct location then the method call shall have no effect.

Parameters:

aDVBService - The coordinates of the DVB service which contains the object carousel. This locator has to point to a DVB service.

aCarouselId - The identifier of the carousel.

Throws:

java.io.InterruptedIOException - The attachment has been aborted.

MPEGDeliveryException - An MPEG error occurred (such as time-out).

ServiceXFRException - The service gateway cannot be loaded in the current service domain. This exception shall not be thrown in this version of the specification.

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

detach()

```
public void detach()
throws NotLoadedException
```

A call to this method is a hint that the applications gives to the MHP to unmount the volume and delete the objects of the service domain. When another application is using objects of the same service domain the method has no effects. When there are no other application using objects of the service domain, a call to this method is a hint that the MHP can free all the resources allocated to this service domain.

After this, the `ServiceDomain` will be in a non-attached state and will behave as if it had just been constructed. Subsequent calls to `detach` shall throw `NotLoadedException`.

Throws:

`NotLoadedException` - is thrown if the ServiceDomain is not attached or if there is no call to `attach` in progress.

getContextData(int)

```
public byte[] getContextData(int context_id)
throws NotLoadedException
```

Returns the context_data_bytes from the service context list of the ServiceDomain's service gateway object.

Parameters:

`context_id` - the context_id whose context_data is to be returned.

Returns:

an array containing the context_data_bytes or null if the context_id specified is not present in the service context list.

Throws:

`NotLoadedException` - if the service domain is not loaded.

getLocator()

```
public org.davic.net.Locator getLocator()
```

Return the locator for this service domain. If this ServiceDomain instance was last attached by specifying a locator then an equivalent locator shall be returned except if the original locator contained extra information that is not necessary to identify the service domain in which case this extra information is removed. If the attach was done with the `attach(locator, int)` signature, the locator is complemented with the component_tag value that the platform has identified during attaching the ServiceDomain. If this ServiceDomain instance was last attached by specifying an NSAP address then the locator shall be generated from that address. If this ServiceDomain has never been attached then null shall be returned.

The syntax of the NSAP address is defined in section titled "LiteOptionsProfileBody" in annex B of the MHP specification. It contains the same fields as the locator syntax specified in the System integration aspects clause. The locator is constructed by taking the fields out of the NSAP address and encoding them in the locator syntax together with the component_tag value that the platform has identified during attaching the ServiceDomain.

Returns:

a locator for this service domain.

Since:

MHP 1.0.1.

getMountPoint()

```
public org.dvb.dsmcc.DSMCCObject getMountPoint()
```

Returns a `DSMCCObject` object describing the top level directory of this `ServiceDomain`. If the ServiceDomain object is not attached then null is returned.

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

Returns:

an instance of org.dvb.dsmcc.DSMCCObject if attached or null otherwise.

Since:

MHP 1.0.1.

getNSAPAddress()

```
public byte[] getNSAPAddress()
throws NotLoadedException
```

This method returns the NSAP address of the `ServiceDomain`.

Returns:

the NSAP address of the `ServiceDomain`.

Throws:

`NotLoadedException` - is thrown if the `ServiceDomain` is not attached.

getURL(Locator)

```
public static java.net.URL getURL(org.davic.net.Locator l)
throws NotLoadedException, InvalidLocatorException, FileNotFoundException
```

Obtain a java.net.URL corresponding to a 'dvb:' locator. If the service domain corresponding to the locator is attached and the file referenced in the locator exists then an instance of `java.net.URL` is returned which can be used to reference this file.

Parameters:

`l` - a locator object encapsulating a 'dvb:' locator which includes a 'dvb_abs_path' element.

Returns:

a `java.net.URL` which can be used to access the file referenced by the 'dvb:' locator.

Throws:

`org.davic.net.InvalidLocatorException` - if the locator is not a valid 'dvb:' locator or does not includes all elements including 'dvb_abs_path' element.

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

`java.io.FileNotFoundException` - if the service domain is attached but the file referenced by the locator does not exist.

isAttached()

```
public boolean isAttached()
```

Return whether this service domain is in the attached or detached state.

Returns:

true if this service domain is in the attached state, otherwise false.

Since:

MHP 1.0.1.

isNetworkConnectionAvailable()

```
public boolean isNetworkConnectionAvailable()
```


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

Return whether the network connection for this service domain is available. This return value is independent of whether the service domain is attached or not. If a service domain is distributed across multiple network connections (e.g. using the optional support for DSMCC over IIOP) then this will reflect the availability of the network connection carrying the object mounted to the mount point.

Returns:

true if the network connection for this service domain is available otherwise false.

Since:

MHP 1.0.1.

org.dvb.dsmcc

ServiceXFRErrorEvent

Declaration

```
public class ServiceXFRErrorEvent extends AsynchronousLoadingEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
              |
              +--org.dvb.dsmcc.ServiceXFRErrorEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

The object requested is available in an alternate ServiceDomain. When an application attempts to asynchronously load an object that has itself a LiteOptionProfileBody IOR or that has a parent directory that has a LiteOptionProfileBody IOR, this event shall be sent to the application. There is no implicit mounting by the implementation of the carousel that actually contains the object. This event is not sent if the Service Domain that actually contains the DSMCCObject is already mounted.

Constructors

ServiceXFRErrorEvent(DSMCCObject, ServiceXFRReference)

```
public ServiceXFRErrorEvent(org.dvb.dsmcc.DSMCCObject o, org.dvb.dsmcc.ServiceXFRReference ref)
```

Creates a ServiceXFRErrorEvent object.

Parameters:

`o` - the DSMCCObject that generated the event.

`ref` - the address of an alternate ServiceDomain where the object can be found.

Methods

getServiceXFR()

```
public org.dvb.dsmcc.ServiceXFRReference getServiceXFR()
```

This method is used to get a reference to the service domain that contains the requested object.

Returns:

The address of an alternate ServiceDomain where the object can be found.

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

getSource()

```
public java.lang.Object getSource()
```

Returns the DSMCCObject that generated the event.

Overrides:

getSource in class AsynchronousLoadingEvent

Returns:

the DSMCCObject that generated the event.

org.dvb.dsmcc

ServiceXFRException

Declaration

```
public class ServiceXFRException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.ServiceXFRException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

A ServiceXFRException is thrown when a DSMCC Object can not be loaded in the current ServiceDomain but is available in an alternate ServiceDomain (i.e. for an object Carousel, the IOR of the object or one of its parent directories contains a Lite Option Profile Body). There is no implicit mounting by the implementation of the carousel that actually contain the object. This exception is not thrown if the Service Domain that actually contains the DSMCCObject is already mounted.

Constructors

ServiceXFRException(byte[], String)

```
public ServiceXFRException(byte[] NSAPAddress, java.lang.String pathName)
```

Creates a ServiceXFRException object.

Parameters:

NSAPAddress - The NSAP Address of a ServiceDomain as defined in ISO/IEC 13818-6 [38].

pathName - pathName of the object in the alternate ServiceDomain.

ServiceXFRException(Locator, int, String)

```
public ServiceXFRException(org.davic.net.Locator aService, int carouselId,
java.lang.String pathName)
```

Creates a ServiceXFRException object.

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

Parameters:

aService - Locator of the Service.

carouselId - Carousel Identifier.

pathName - pathName of the object in the alternate ServiceDomain.

Methods

getServiceXFR()

```
public org.dvb.dsmcc.ServiceXFRReference getServiceXFR()
```

This method is used to get the alternate ServiceDomain which contains the object requested.

Returns:

the address of an alternate ServiceDomain where the object can be found.

org.dvb.dsmcc ServiceXFRReference

```
Declaration
public class ServiceXFRReference

java.lang.Object
  |
  +--org.dvb.dsmcc.ServiceXFRReference
```

Description

A ServiceXFRReference object is used when a DSMCC Object can not be loaded in the current ServiceDomain but is available in an alternate ServiceDomain. Instances of this class are just containers. The parameters passed are merely stored and returned by the access methods. It is the responsibility of the platform when generating instances to use correct values.

Constructors

ServiceXFRReference(byte[], String)

```
public ServiceXFRReference(byte[] nsapAddress, java.lang.String pathName)
```

Creates a ServiceXFRReference object.

Parameters:

nsapAddress - The NSAP Address of a ServiceDomain as defined in ISO/IEC 13818-6 [38].

pathName - pathName of the object in the alternate ServiceDomain.

ServiceXFRReference(Locator, int, String)

```
public ServiceXFRReference(org.davic.net.Locator serviceLocator, int carouselId,
java.lang.String pathName)
```

Creates a ServiceXFRReference object.

Parameters:

serviceLocator - Locator of the Service.

carouselId - Carousel Identifier.

pathName - pathName of the object in the alternate ServiceDomain.

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

Methods

getCarouselId()

```
public int getCarouselId()
```

This method returns the carousel identifier. If the object was constructed using the constructor which includes a carousel ID or if it was constructed using the constructor which includes an NSAP address and that NSAP address contains a carouselID then this method shall return that carousel ID otherwise this method shall return -1.

Returns:

the carousel identifier or -1.

getLocator()

```
public org.davic.net.Locator getLocator()
```

This method returns the Locator of the Service for an Object Carousel.

Returns:

the Locator of the Service for an Object Carousel. This method returns null, if the ServiceDomain is not associated with an Object Carousel. In this case the NSAP address must be used instead.

getNSAPAddress()

```
public byte[] getNSAPAddress()
```

This method returns the NSAP Address of a ServiceDomain as defined in ISO/IEC 13818-6 [38]. If the object was constructed using an NSAP address then this method shall return the NSAP address passed into the constructor. If the object was constructed with a locator and a carouselID then this method shall return an NSAP address derived from this information when locator is an instance of org.davic.net.dvb.DVBLocator. Otherwise this method shall return null.

Returns:

the NSAP Address of a ServiceDomain as defined in ISO/IEC 13818-6 [38] or null.

getPathName()

```
public java.lang.String getPathName()
```

This method returns the pathname of the object in the alternate ServiceDomain.

Returns:

the pathname of the object in the alternate ServiceDomain.

org.dvb.dsmcc

StreamEvent

Declaration

```
public class StreamEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.StreamEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```


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

Description

This class describes a Stream event which is used to synchronize an application with an MPEG Stream.

NOTE: The NPT mechanism and scheduled stream events that depend on it are known to be vulnerable to disruption in many digital TV distribution networks. Existing deployed network equipment that re-generates the STC is unlikely to be aware of NPT and hence will not make the necessary corresponding modification to STC values inside NPT reference descriptors. This may cause stream events scheduled against NPT to fire at the wrong time or to never fire at all. Applications should only use scheduled stream events with NPT where they are confident that the network where they are to be used does not have this problem. DVB Timeline, DSM-CC "do it now" events and events carried within DVB synchronised auxiliary data offer more reliable alternatives to NPT.

Constructors

StreamEvent(DSMCCStreamEvent, long, String, int, byte[])

```
public StreamEvent(org.dvb.dsmcc.DSMCCStreamEvent source, long npt, java.lang.String name,
int eventId, byte[] eventData)
```

Creates a StreamEvent object.

Parameters:

`source` - The `DSMCCStreamEvent` that has generated the event.

`npt` - The value of the timeline when the event is triggered. For NPT based timelines, this value is equal to the field eventNPT in the DSMCC StreamEventDescriptor except where the event is a "do it now" event in which case the value -1 is returned (as the value of NPT may not be meaningful). For DVB timelines, this value is the to-be-completed

`name` - The name of this event. The list of event names is located in the DSMCC StreamEvent object. This list is returned by the method `DSMCCStreamEvent.getEventList`.

`eventId` - The eventId of this event. The list of event IDs is located in the DSMCC StreamEvent object.

`eventData` - The application specific data found in the DSMCC StreamEventDescriptor.

Methods

getEventData()

```
public byte[] getEventData()
```

This method is used to retrieve the private data associated with the event. For events signalled by a DVB synchronised event descriptor, these are the bytes carried in the synchronised_event_data_byte field. For events signalled by a DSMCC stream event descriptor, these are the contents of the privateDataByte field.

Returns:

The private data associated with the event.

getEventId()

```
public int getEventId()
```

This method is used to get the identifier of the StreamEvent.

Returns:

The identifier of the StreamEvent.

getEventName()

```
public java.lang.String getEventName()
```

This method is used to get the name of the StreamEvent.

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

Returns:

the name of the StreamEvent.

getEventNPT()

```
public long getEventNPT()
```

This method is used to get the value of the timeline when the event was triggered.

Returns:

The value of the timeline in milliseconds.

getSource()

```
public java.lang.Object getSource()
```

This method returns the DSMCCStreamEvent that generated the event.

Overrides:

`getSource` in class `EventObject`

Returns:

the DSMCCStreamEvent that generated the event.

org.dvb.dsmcc StreamEventListener

Declaration

```
public interface StreamEventListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

Objects that implement the StreamEventListener interface can receive StreamEvent event.

Methods

receiveStreamEvent(StreamEvent)

```
public void receiveStreamEvent(org.dvb.dsmcc.StreamEvent e)
```

Send a StreamEvent to the StreamEventListener.

Parameters:

`e` - the StreamEvent event.

org.dvb.dsmcc SuccessEvent

Declaration

```
public class SuccessEvent extends AsynchronousLoadingEvent
```


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

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.dsmcc.AsynchronousLoadingEvent
              |
              +--org.dvb.dsmcc.SuccessEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This event indicates that the asynchronous loading was successful.

Constructors

SuccessEvent(DSMCCObject)

```
public SuccessEvent(org.dvb.dsmcc.DSMCCObject o)
```

Creates a `SuccessEvent` object.

Parameters:

`o` - the `DSMCCObject` which was successfully loaded.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the `DSMCCObject` which was successfully loaded.

Overrides:

`getSource` in class `AsynchronousLoadingEvent`

Returns:

the loaded `DSMCCObject`

org.dvb.dsmcc

UnknownEventException

Declaration

```
public class UnknownEventException extends DSMCCException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.io.IOException
                    |
                    +--org.dvb.dsmcc.DSMCCException
                          |
                          +--org.dvb.dsmcc.UnknownEventException
```

All Implemented Interfaces:

```
java.io.Serializable
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 531 of 861

Description

The `UnknownEventException` is thrown when a method tries to access to an unknown event. This exception may get thrown because the event in question is not being signalled yet. It does not indicate that the event is permanently unavailable. Applications may choose to attempt to subscribe to the event again at a later point in time in the expectation that the event has become available since the previous attempt.

Constructors

UnknownEventException()

```
public UnknownEventException()
```

Construct an UnknownEventException with no detail message.

UnknownEventException(String)

```
public UnknownEventException(java.lang.String s)
```

Construct an UnknownEventException with the specified detail message.

Parameters:

`s` - the detail message.

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

Annex Q (normative): Datagram socket buffer control

Package

org.dvb.net

Description

Provides general networking features not found elsewhere.

Class Summary	
Classes	
DatagramSocketBufferControl	This class provides additional control over buffering for DatagramSockets.

org.dvb.net

DatagramSocketBufferControl

Declaration

```
public class DatagramSocketBufferControl

java.lang.Object
  |
  +--org.dvb.net.DatagramSocketBufferControl
```

Description

This class provides additional control over buffering for DatagramSockets.

Methods

getReceiveBufferSize(DatagramSocket)

```
public static int getReceiveBufferSize(java.net.DatagramSocket d)
throws SocketException
```

Get value of the SO_RCVBUF option for this socket, that is the buffer size used by the platform for input on the this Socket. The value returned need not be the value previously set by setReceiveBufferSize (if any).

Parameters:

d - The DatagramSocket for which to query the receive buffer size.

Returns:

The size of the receive buffer, in bytes.

Throws:

java.net.SocketException - - If there is an error when querying the SO_RCVBUF option.

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

setReceiveBufferSize(DatagramSocket, int)

```
public static void setReceiveBufferSize(java.net.DatagramSocket d, int size)
throws SocketException
```

Sets the SO_RCVBUF option to the specified value for this DatagramSocket. The SO_RCVBUF option is used by the platform's networking code as a hint for the size to use when allocating the underlying network I/O buffers.

Increasing buffer size can increase the performance of network I/O for high-volume connection, while decreasing it can help reduce the backlog of incoming data. For UDP, this sets the buffer size for received packets.

Because SO_RCVBUF is a hint, applications that want to verify what size the buffers were set to should call get-ReceiveBufferSize. This method shall throw IllegalArgumentException - if size is 0 or is negative.

Parameters:

`d` - The DatagramSocket for which to change the receive buffer size.

`size` - The requested size of the receive buffer, in bytes.

Throws:

`java.net.SocketException` - - If there is an error when setting the SO_RCVBUF option.

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

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

Package

org.dvb.net.rc

Description

Provides session management for bi-directional IP connections which are session based from the point of view of an application. The best example of this is a conventional modem.

Class Summary	
Interfaces	
`ConnectionListener`	This interface should be implemented by objects wishing to be notified about the connection status of a `ConnectionRCInterface`.
Classes	
`ConnectionDroppedEvent`	ConnectionDroppedEvent - An event generated after an attempt to setup a connection for a `ConnectionRCInterface` fails due to the connection being dropped by the target.
`ConnectionEstablishedEvent`	ConnectionEstablishedEvent - An event generated after a connection is established for a `ConnectionRCInterface`.
`ConnectionFailedEvent`	ConnectionFailedEvent - An event generated after an attempt to setup a connection for a `ConnectionRCInterface` fails.
`ConnectionParameters`	This class encapsulates the parameters needed to specify the target of a connection.
`ConnectionRCEvent`	ConnectionRCEvent - the base class for events related to connection oriented return channels.
`ConnectionRCInterface`	This class models a connection based return channel network interface for use in receiving and transmitting IP packets over a return channel.
`ConnectionTerminatedEvent`	ConnectionTerminatedEvent - An event generated after a connected `ConnectionRCInterface` is disconnected.
`NoDialToneEvent`	NoDialToneEvent - An event generated after an attempt to setup a connection for a `ConnectionRCInterface` of `TYPE_PSTN` fails due to there not being a dial tone on the return channel concerned.
`RCInterface`	This class models a return channel network interface for use in receiving and transmitting IP packets over a logical return channel.
`RCInterfaceManager`	This class is the factory and manager for all return channel interfaces in the system.
`RCInterfaceReleasedEvent`	This event informs an application that a `RCInterface` has been released by an application or other entity in the system.
`RCInterfaceReservedEvent`	This event informs an application that a `RCInterface` has been reserved by an application or other entity in the system.
`RCPermission`	This class is for return channel set-up permissions.
`TargetBusyEvent`	TargetBusyEvent - An event generated after an attempt to setup a connection for a `ConnectionRCInterface` fails due to the target of the connection being busy.
Exceptions	
`IncompleteTargetException`	Thrown when the target for a connection is incompletely specified.
`PermissionDeniedException`	Thrown when an application calls a method which it does not have permission to call at that time.

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

org.dvb.net.rc ConnectionDroppedEvent

Declaration

```
public class ConnectionDroppedEvent extends ConnectionFailedEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.net.rc.ConnectionRCEvent
              |
              +--org.dvb.net.rc.ConnectionFailedEvent
                    |
                    +--org.dvb.net.rc.ConnectionDroppedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

ConnectionDroppedEvent - An event generated after an attempt to setup a connection for a `ConnectionRCInterface` fails due to the connection being dropped by the target. For a ConnectionRCInterface of type `TYPE_PSTN`, this includes the following conditions:

- Connection was dropped after ringing but before it was answered.
- Connection rang but was never answered and timed-out.
- Connection was dropped after answering but before the IP connection was established.
- Connection was answered but the IP connection was never established and timed-out.

Constructors

ConnectionDroppedEvent(Object)

```
public ConnectionDroppedEvent(java.lang.Object source)
```

Construct an event.

Parameters:

`source` - the `ConnectionRCInterface` whose connection attempt failed.

org.dvb.net.rc ConnectionEstablishedEvent

Declaration

```
public class ConnectionEstablishedEvent extends ConnectionRCEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.net.rc.ConnectionRCEvent
              |
              +--org.dvb.net.rc.ConnectionEstablishedEvent
```


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

All Implemented Interfaces:

`java.io.Serializable`

Description

ConnectionEstablishedEvent - An event generated after a connection is established for a `ConnectionRCInterface`.

Constructors

ConnectionEstablishedEvent(Object)

`public` **`ConnectionEstablishedEvent`**`(java.lang.Object source)`

Construct an event.

Parameters:

`source` - the `ConnectionRCInterface` whose connection was established.

org.dvb.net.rc

ConnectionFailedEvent

Declaration

`public class` **`ConnectionFailedEvent`** `extends ConnectionRCEvent`

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.net.rc.ConnectionRCEvent
              |
              +--org.dvb.net.rc.ConnectionFailedEvent
```

All Implemented Interfaces:

`java.io.Serializable`

Direct Known Subclasses:

`ConnectionDroppedEvent, NoDialToneEvent, TargetBusyEvent`

Description

ConnectionFailedEvent - An event generated after an attempt to setup a connection for a `ConnectionRCInterface` fails.

Constructors

ConnectionFailedEvent(Object)

`public` **`ConnectionFailedEvent`**`(java.lang.Object source)`

Construct an event.

Parameters:

`source` - the `ConnectionRCInterface` whose connection attempt failed.

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

org.dvb.net.rc ConnectionListener

Declaration

```
public interface ConnectionListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

This interface should be implemented by objects wishing to be notified about the connection status of a `ConnectionRCInterface`.

Methods

connectionChanged(ConnectionRCEvent)

```
public void connectionChanged(org.dvb.net.rc.ConnectionRCEvent e)
```

This method is called to report events related to the setup and termination of return channel interface connections.

Parameters:

`e` - the event which happened.

org.dvb.net.rc ConnectionParameters

Declaration

```
public class ConnectionParameters
```

```
java.lang.Object
  |
  +--org.dvb.net.rc.ConnectionParameters
```

Description

This class encapsulates the parameters needed to specify the target of a connection.

Constructors

ConnectionParameters(String, String, String)

```
public ConnectionParameters(java.lang.String number, java.lang.String username,
java.lang.String password)
```

Construct a set of connection parameters. Details of the DNS server to use are supplied by the server.

Parameters:

`number` - the target of the connection, e.g. a phone number.

`username` - the username to use in connection setup.

`password` - the password to use in connection setup.

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

ConnectionParameters(String, String, String, InetAddress[])

```
public ConnectionParameters(java.lang.String number, java.lang.String username,
java.lang.String password, java.net.InetAddress[] dns)
```

Construct a set of connection parameters.

Parameters:

`number` - the target of the connection, e.g. a phone number.

`username` - the username to use in connection setup.

`password` - the password to use in connection setup.

`dns` - the list of DNS servers to try before reporting failure. The order in which they are interrogated is not specified. Once one result has been obtained, there is no requirement to try others.

Methods

getDNSServer()

```
public java.net.InetAddress[] getDNSServer()
```

Return the addresses of the DNS servers to use for the connection.

Returns:

return the addresses of the DNS servers passed into the constructor of the instance or null if none was provided.

getPassword()

```
public java.lang.String getPassword()
```

Return the password used in establishing this connection The value returned shall be the one passed into the constructor of this instance.

Returns:

the password used in establishing this connection.

getTarget()

```
public java.lang.String getTarget()
```

Return the target of this connection for example a phone number. The value returned shall be the one passed into the constructor of this instance.

Returns:

the target of the connection.

getUsername()

```
public java.lang.String getUsername()
```

Return the username used in establishing this connection The value returned shall be the one passed into the constructor of this instance.

Returns:

the username used in establishing the connection.

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

org.dvb.net.rc
ConnectionRCEvent

Declaration

```
public class ConnectionRCEvent extends java.util.EventObject

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.net.rc.ConnectionRCEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Direct Known Subclasses:

```
ConnectionEstablishedEvent, ConnectionFailedEvent, ConnectionTerminatedEvent
```

Description

ConnectionRCEvent - the base class for events related to connection oriented return channels.

Constructors

ConnectionRCEvent(Object)

```
public ConnectionRCEvent(java.lang.Object source)
```

Construct an event.

Parameters:

`source` - the `ConnectionRCInterface` for which the event was generated.

org.dvb.net.rc
ConnectionRCInterface

Declaration

```
public class ConnectionRCInterface extends RCInterface implements org.davic.resources.ResourceProxy

java.lang.Object
  |
  +--org.dvb.net.rc.RCInterface
        |
        +--org.dvb.net.rc.ConnectionRCInterface
```

All Implemented Interfaces:

```
org.davic.resources.ResourceProxy
```

Description

This class models a connection based return channel network interface for use in receiving and transmitting IP packets over a return channel. Targets for connections are specified as strings including the number to dial. These strings can only include either numbers or a "+" character (as the first character only).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 540 of 861

When a `ConnectionRCInterface` instance is first obtained by an application, the current target shall be set to the default. Applications which wish to use a non-default target need to set this target before attempting to reserve the `ConnectionRCInterface`. This is because if the application does not have the permission to use the default target, the `reserve()` method is required throw a `SecurityException`.

Constructors

ConnectionRCInterface()

```
protected ConnectionRCInterface()
```

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

addConnectionListener(ConnectionListener)

```
public void addConnectionListener(org.dvb.net.rc.ConnectionListener l)
```

Add a listener for events related to connections of this interface.

Parameters:

`l` - the listener for the connection related events.

connect()

```
public void connect()
throws IOException, PermissionDeniedException
```

Connect this return channel to the current target. If this ResourceProxy does not have the underlying resource reserved then a PermissionDeniedException will be thrown. Where the underlying resource is reserved but at the time the method is called, it is known that connection is impossible then an IOException will be thrown. Apart from this, this method is asynchronous and completion or failure is reported through the event listener on this class. If a connection is already established when this method is called then the method call shall have no effect.

Throws:

`PermissionDeniedException` - if this application does not own the resource.

`java.io.IOException` - if connection is known to be impossible at the time when the method is called.

disconnect()

```
public void disconnect()
throws PermissionDeniedException
```

Disconnect this return channel from the current target. This method is asynchronous and completion is reported through the event listener on this class. This method does not release the underlying resource from the ResourceProxy. If no connection is established then this method shall have no effect.

Throws:

`PermissionDeniedException` - if this application does not own the resource.

getClient()

```
public org.davic.resources.ResourceClient getClient()
```

Return the object which asked to be notified about withdrawal of the underlying resource. This is the object provided as the first parameter to the last call to the reserve method on this object. If this object does not have the underlying resource reserved then null is returned.

Specified By:

`getClient` in interface `ResourceProxy`

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 541 of 861

Returns:

the object which asked to be notified about withdrawal of the underlying physical resource from this resource proxy or null.

getConnectedTime()

```
public int getConnectedTime()
```

Return the time an interface has been connected.

Returns:

the time in seconds for which this interface has been connected or -1 if the device is not connected.

getCurrentTarget()

```
public org.dvb.net.rc.ConnectionParameters getCurrentTarget()
throws IncompleteTargetException
```

Get the current target for connections.

If this ConnectionRCInterface is connected then this method shall return the target to which the connection was made. If this ConnectionRCInterface is not connected then this method shall return the last target set by the setTarget method (if any) otherwise the default.

This returns either the default target or the last target set by this application calling the setTarget method on this instance before the connection was established. This applies regardless of whether the connection was established by another GEM application or if some of the connection parameters have been supplied by the server.

Returns:

the current set of connection target parameters.

Throws:

`IncompleteTargetException` - if the current target is not completely configured.

`java.lang.SecurityException` - if the application is not allowed to read the current target as defined by the security policy of the platform.

getSetupTimeEstimate()

```
public float getSetupTimeEstimate()
```

Obtain an estimate of the setup time for a successful connection for this interface in seconds.

Returns:

an estimate of the setup time for a successful connection for this interface in seconds.

isConnected()

```
public boolean isConnected()
```

Check if this interface is connected. Connected means able to receive and transmit packets.

Returns:

true if the interface is connected, otherwise false.

release()

```
public void release()
```

Release the right to control this return channel interface. If this object does not have the right to control this return channel interface then this method shall have no effect.

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

removeConnectionListener(ConnectionListener)

```
public void removeConnectionListener(org.dvb.net.rc.ConnectionListener l)
```

Remove a listener for events related to connections of this interface. If the listener specified is not currently receiving these events then this method has no effect.

Parameters:

`l` - the listener for the connection related events.

reserve(ResourceClient, Object)

```
public void reserve(org.davic.resources.ResourceClient c, java.lang.Object requestData)
throws PermissionDeniedException
```

Request the right to control this return channel interface. If the right to control the return channel interface has already been reserved then this method shall have no effect.

The details of the current connection target shall be obtained from the `ConnectionParameters` instance which is the current target during the call to this method. Hence changes to that `ConnectionParameters` instance before a call to this method shall be taken account of during the method call. Changes after the call to this method shall have no effect on that connection.

Parameters:

`c` - the object to be notified when resources are removed.

`requestData` - Used by the Resource Notification API in the requestRelease method of the ResourceClient interface. The usage of this parameter is optional and a null reference may be supplied.

Throws:

`PermissionDeniedException` - if this interface cannot be reserved.

`java.lang.SecurityException` - if the application is denied access to the resource by security policy.

setTarget(ConnectionParameters)

```
public void setTarget(org.dvb.net.rc.ConnectionParameters target)
throws IncompleteTargetException, PermissionDeniedException
```

Set a non-default target for connections.

If this method is called for a ConnectionRCInterface which is connected then successful calls to this method shall not interrupt that connection. The newly set target shall just be stored until either the next time a connection is established with that ConnectionRCInterface instance or until a subsequent call to setTarget on that ConnectionRCInterface.

The details of the current connection target shall be obtained from the newly set target during the call to this method. Changes to that instance after the call to this method shall have no effect on the ConnectionRCInterface unless/until setTarget is called again for that instance or it is released and reserved again.

Parameters:

`target` - the new set of connection target parameters.

Throws:

`IncompleteTargetException` - if the application owns the resource but the target is not completely specified.

`PermissionDeniedException` - this exception shall never be thrown.

`java.lang.SecurityException` - if the application is not allowed to modify the target as defined by the security policy of the platform.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 543 of 861

setTargetToDefault()

```
public void setTargetToDefault()
throws PermissionDeniedException
```

Set the target for connections to the default.

Throws:

`PermissionDeniedException` - if this application does not own the resource.

`java.lang.SecurityException` - if the application is not allowed to connect to the default target.

org.dvb.net.rc ConnectionTerminatedEvent

Declaration

```
public class ConnectionTerminatedEvent extends ConnectionRCEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.net.rc.ConnectionRCEvent
              |
              +--org.dvb.net.rc.ConnectionTerminatedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

ConnectionTerminatedEvent - An event generated after a connected `ConnectionRCInterface` is disconnected.

Constructors

ConnectionTerminatedEvent(Object)

```
public ConnectionTerminatedEvent(java.lang.Object source)
```

Construct an event.

Parameters:

`source` - the `ConnectionRCInterface` whose status changed.

org.dvb.net.rc IncompleteTargetException

Declaration

```
public class IncompleteTargetException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.net.rc.IncompleteTargetException
```


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

All Implemented Interfaces:

`java.io.Serializable`

Description

Thrown when the target for a connection is incompletely specified. This is thrown either when the default connection target is incompletely defined in the device or when an application provides an incompletely defined connection target to the device or when the connection target is badly formed, e.g. includes illegal characters in a number parameter.

Constructors

IncompleteTargetException()

```
public IncompleteTargetException()
```

Default constructor for the exception

IncompleteTargetException(String)

```
public IncompleteTargetException(java.lang.String reason)
```

Constructor for the exception with a specified reason.

Parameters:

`reason` - the reason why the exception was raised.

org.dvb.net.rc

NoDialToneEvent

Declaration

```
public class NoDialToneEvent extends ConnectionFailedEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.net.rc.ConnectionRCEvent
              |
              +--org.dvb.net.rc.ConnectionFailedEvent
                    |
                    +--org.dvb.net.rc.NoDialToneEvent
```

All Implemented Interfaces:

`java.io.Serializable`

Description

NoDialToneEvent - An event generated after an attempt to setup a connection for a `ConnectionRCInterface` of `TYPE_PSTN` fails due to there not being a dial tone on the return channel concerned.

Constructors

NoDialToneEvent(Object)

```
public NoDialToneEvent(java.lang.Object source)
```

Construct an event.

Parameters:

`source` - the `ConnectionRCInterface` whose connection attempt failed.

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

org.dvb.net.rc PermissionDeniedException

Declaration

```
public class PermissionDeniedException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.net.rc.PermissionDeniedException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown when an application calls a method which it does not have permission to call at that time.

Constructors

PermissionDeniedException()

```
public PermissionDeniedException()
```

Default constructor for the exception.

PermissionDeniedException(String)

```
public PermissionDeniedException(java.lang.String reason)
```

Constructor for the exception with a specified reason.

Parameters:

`reason` - the reason why the exception was raised.

org.dvb.net.rc RCInterface

Declaration

```
public class RCInterface

java.lang.Object
  |
  +--org.dvb.net.rc.RCInterface
```

Direct Known Subclasses:

```
ConnectionRCInterface
```

Description

This class models a return channel network interface for use in receiving and transmitting IP packets over a logical return channel. This can include real analog modems, cable return channel and all the other options allowed by the relevant DVB specification. This class does not model any concept of connection. Hence interfaces represented by this class and not by a sub-class of it are permanently connected.

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

Fields

TYPE_CATV

```
public static final int TYPE_CATV
```

Constant to indicate a CATV return channel.

TYPE_DECT

```
public static final int TYPE_DECT
```

Constant to indicate a DECT return channel.

TYPE_ISDN

```
public static final int TYPE_ISDN
```

Constant to indicate an ISDN return channel.

TYPE_LMDS

```
public static final int TYPE_LMDS
```

Constant to indicate a LMDS return channel.

TYPE_MATV

```
public static final int TYPE_MATV
```

Constant to indicate a MATV return channel.

TYPE_OTHER

```
public static final int TYPE_OTHER
```

Constant to indicate all other return channel technologies not having a suitable defined constant in this class.

NOTE: DVB does not intend to add future constants to this list for future return channel technologies. These should be represented as TYPE_OTHER.

TYPE_PSTN

```
public static final int TYPE_PSTN
```

Constant to indicate a PSTN return channel.

TYPE_RCS

```
public static final int TYPE_RCS
```

Constant to indicate a DVB-RCS return channel.

TYPE_UNKNOWN

```
public static final int TYPE_UNKNOWN
```

Constant to indicate an unknown return channel technology. There is an intermediate physical interface between the GEM terminal and the return channel device. This return value gives no information about whether the return channel is connection oriented or connectionless.

Constructors

RCInterface()

```
protected RCInterface()
```

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.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 547 of 861

Methods

getDataRate()

```
public int getDataRate()
```

Return the maximum data rate of the connection over the immediate access network to which this network interface is connected. For asymmetric connections, the data rate coming into the GEM terminal shall be returned. For connection oriented interfaces which are not currently connected, the value returned shall be that of the last connection established where that information is available. Where that information is not available, (e.g. where no connection has been established since a GEM terminal was power cycled), -1 shall be returned.

Returns:

a data rate in KBaud or -1 where this is not available.

Since:

MHP 1.0.1.

getType()

```
public int getType()
```

Return the type of return channel represented by this object. Note, applications wishing to discover whether a return channel interface is connection oriented or not are recommended to test whether an object is an instance of `ConnectionRCInterface` or not. A non-connection oriented interface really means a permanently connected return channel.

Returns:

the type of return channel represented by this object encoded as one of the constants defined in this class.

org.dvb.net.rc

RCInterfaceManager

Declaration

```
public class RCInterfaceManager implements org.davic.resources.ResourceServer
```

```
java.lang.Object
  |
  +--org.dvb.net.rc.RCInterfaceManager
```

All Implemented Interfaces:

```
org.davic.resources.ResourceServer
```

Description

This class is the factory and manager for all return channel interfaces in the system. The methods on this class which return instances of the `RCInterface` will only return new instances of that class under the following conditions:

- on the first occasion an instance needs to be returned to a particular application for a particular interface;
- when new return channel interfaces are added to the system.

Methods

addResourceStatusEventListener(ResourceStatusListener)

```
public void addResourceStatusEventListener(org.davic.resources.ResourceStatusListener listener)
```

This method informs a resource server that a particular object should be informed of changes in the state of the resources managed by that server.

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

Specified By:

`addResourceStatusEventListener` in interface `ResourceServer`

Parameters:

`listener` - the object to be informed of state changes.

getInstance()

```
public static org.dvb.net.rc.RCInterfaceManager getInstance()
```

Factory method to obtain a manager. The RCInterfaceManager is either a singleton for each GEM application or a singleton for the GEM terminal.

Returns:

an instance of an RCInterfaceManager.

getInterface(InetAddress)

```
public org.dvb.net.rc.RCInterface getInterface(java.net.InetAddress addr)
```

Return the interface which will be used when connecting to a particular host. Null is returned if this is not known when the method is called.

Parameters:

`addr` - the IP address of the host to connect to.

Returns:

the interface which will be used or null if this is not known.

getInterface(Socket)

```
public org.dvb.net.rc.RCInterface getInterface(java.net.Socket s)
```

Return the interface which is used for a particular socket.

Parameters:

`s` - the socket to use.

Returns:

the interface which is used or null if the socket is not connected.

getInterface(URLConnection)

```
public org.dvb.net.rc.RCInterface getInterface(java.net.URLConnection u)
```

Return the interface which is used for a particular URLConnection.

Parameters:

`u` - the URLConnection to use.

Returns:

the interface which is used or null if the URLConnection is not connected.

getInterfaces()

```
public org.dvb.net.rc.RCInterface[] getInterfaces()
```

Factory method to return a list of all return channel interfaces visible to this application. The number of entries in the array will exactly match the number of return channel interfaces visible to the application. Null is returned if no interfaces are visible to this application.

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

Returns:

an array of available return channel interfaces.

removeResourceStatusEventListener(ResourceStatusListener)

```
public void removeResourceStatusEventListener(org.davic.resources.ResourceStatusListener listener)
```

This method informs a resource server that a particular object is no longer interested in being informed about changes in state of resources managed by that server. If the object had not registered its interest initially then this method has no effect.

Specified By:

`removeResourceStatusEventListener` in interface `ResourceServer`

Parameters:

`listener` - the object which is no longer interested.

org.dvb.net.rc

RCInterfaceReleasedEvent

Declaration

```
public class RCInterfaceReleasedEvent extends org.davic.resources.ResourceStatusEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.davic.resources.ResourceStatusEvent
              |
              +--org.dvb.net.rc.RCInterfaceReleasedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This event informs an application that a `RCInterface` has been released by an application or other entity in the system. It is generated when an application which had successfully reserved a `RCInterface` calls the `Connection-RCInterface.release` method. It will also be generated if any other entities in the system own such an interface and then release that interface in such a way that it could then become available to applications using this API.

Constructors

RCInterfaceReleasedEvent(Object)

```
public RCInterfaceReleasedEvent(java.lang.Object bg)
```

Constructor for the event.

Parameters:

`bg` - the `RCInterface` which has been released.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the device that has been released.

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

Overrides:

getSource in class ResourceStatusEvent

Returns:

the RCInterface object representing the interface that has been released.

org.dvb.net.rc

RCInterfaceReservedEvent

Declaration

```
public class RCInterfaceReservedEvent extends org.davic.resources.ResourceStatusEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.davic.resources.ResourceStatusEvent
              |
              +--org.dvb.net.rc.RCInterfaceReservedEvent
```

All Implemented Interfaces:

java.io.Serializable

Description

This event informs an application that a RCInterface has been reserved by an application or other entity in the system. It is generated when an application successfully reserves a RCInterface. It will also be generated if any other entities in the system reserve such an interface with the effect of something which was visible to applications using this API becoming unavailable.

Constructors

RCInterfaceReservedEvent(Object)

```
public RCInterfaceReservedEvent(java.lang.Object bg)
```

Constructor for the event.

Parameters:

bg - the RCInterface representing the device which has been reserved.

Methods

getSource()

```
public java.lang.Object getSource()
```

Returns the device that has been reserved.

Overrides:

getSource in class ResourceStatusEvent

Returns:

an RCInterface representing the device that has been reserved.

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

org.dvb.net.rc

RCPermission

Declaration

```
public class RCPermission extends java.security.BasicPermission

java.lang.Object
  |
  +--java.security.Permission
        |
        +--java.security.BasicPermission
              |
              +--org.dvb.net.rc.RCPermission
```

All Implemented Interfaces:

```
java.security.Guard, java.io.Serializable
```

Description

This class is for return channel set-up permissions. An RCPermission contains a name, but no actions list.

The permission name can be "target:default", which indicates the permission to use the default connection parameters.

The permission name can also be "target:<phone number>", which indicates the permission to use the specified phone number in the connection set-up (ConnectionRCInterface.setTarget(ConnectionParameters) method).

A wildcard may be used at the end of the permission name. In that case, all phone numbers starting with the number before the wildcard are included in the permission. A "+" may be used at the start of the phone number to indicate a phone number including the international country code.

EXAMPLES:

- target:0206234342 (Permission to dial the specified phone number).
- target:020* (Permission to dial phone numbers starting with 020).
- target:* (Permission to dial all phone numbers, including the default).

NOTE: ConnectionRCInterface.reserve(ResourceClient, Object) will throw a SecurityException if the application is not allowed to set-up a connection over the return channel at all (i.e., there is no valid target allowed).

Constructors

RCPermission(String)

```
public RCPermission(java.lang.String name)
```

Creates a new RCPermission with the specified name. The name is the symbolic name of the RCPermission.

Parameters:

`name` - the name of the RCPermission.

RCPermission(String, String)

```
public RCPermission(java.lang.String name, java.lang.String actions)
```

Creates a new RCPermission object with the specified name. The name is the symbolic name of the RCPermission, and the actions String is unused and should be null. This constructor exists for use by the Policy object to instantiate new Permission objects.

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

Parameters:

`name` - the name of the RCPermission.

`actions` - should be null.

Methods

implies(Permission)

```
public boolean implies(java.security.Permission p)
```

Checks if this RCPermission "implies" the specified Permission.

More specifically, this returns true if and only if:

- p is an instance of RCPermission; and
- p's name is implied by the name of this permission, as described by the wildcarding rules specified in the the description of this class.

Overrides:

`implies` in class `BasicPermission`.

Parameters:

`p` - The Permission to check against.

Returns:

true if the specified Permission is implied by this object; false otherwise.

org.dvb.net.rc

TargetBusyEvent

Declaration

```
public class TargetBusyEvent extends ConnectionFailedEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.net.rc.ConnectionRCEvent
              |
              +--org.dvb.net.rc.ConnectionFailedEvent
                    |
                    +--org.dvb.net.rc.TargetBusyEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

TargetBusyEvent - An event generated after an attempt to setup a connection for a `ConnectionRCInterface` fails due to the target of the connection being busy.

Constructors

TargetBusyEvent(Object)

```
public TargetBusyEvent(java.lang.Object source)
```

Construct an event.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 553 of 861

Parameters:

`source` - the `ConnectionRCInterface` whose connection attempt failed.

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

Annex S (normative): Application listing and launching

Package

org.dvb.application

Description

Provides access to lists of applications which are available in this context and the ability to launch those applications.

Class Summary	
Interfaces	
AppAttributes	The AppAttributes class is a mapping of various information about a registered application.
AppProxy	An AppProxy Object is a proxy to an application.
AppsDatabaseEventListener	The AppsDatabaseListener class allows an application to monitor the application database so that it can keep an up to date interface without polling the state.
AppStateChangeEventListener	The AppStateChangeEventListener class allows a launcher application to keep track of applications it launches or other applications running as part of the same service.
DVBJProxy	A DVBJProxy Object is a proxy to a DVBJ application.
Classes	
AppIcon	The AppIcon encapsulates the information concerning the icon attached to the application
AppID	The AppID is a representation of the unique identifier for applications.
AppsControlPermission	This class represents a Permission to control the lifecycle of another application.
AppsDatabase	The AppsDatabase is an abstract view of the currently available applications.
AppsDatabaseEvent	The AppsDatabaseEvent class indicates either an entry in the application database has changed, or so many changes have occurred.
AppsDatabaseFilter	Abstract class for the filters.
AppStateChangeEvent	The AppStateChangeEvent class indicates a state transition of the application.
CurrentServiceFilter	Instances of CurrentServiceFilter are used to set a filter on the list of applications that are retrieved from the AppsDatabase (See methods getApps-Attributes and getAppsIDs).
RunningApplicationsFilter	Instances of RunningApplicationsFilter are used to set a filter on the list of applications that are retrieved from the AppsDatabase (See methods get-AppsAttributes and getAppsIDs).
Exceptions	
IllegalProfileParameterException	The IllegalProfileParameter exception is thrown if the application attempts to ask for a version number for a profile not specified for the application.
LanguageNotAvailableException	The LanguageNotAvailableException exception is thrown if the application asks for the name of an application in a language not signalled in the AIT.

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

org.dvb.application

AppAttributes

Declaration

```
public interface AppAttributes
```

All Known Subinterfaces:

```
org.dvb.application.storage.ExtendedAppAttributes
```

Description

The `AppAttributes` class is a mapping of various information about a registered application. For applications which are signalled in an AIT, the mapping between the values returned by methods in this class and the fields and descriptors of the AIT shall be as specified in the main body of the present document.

Instances of objects implementing this interface are immutable and populated before the instance is first returned to an application.

Since:

MHP1.0.

Fields

DVB_HTML_application

```
public static final int DVB_HTML_application
```

The DVB registered value for all DVB-HTML applications.

DVB_J_application

```
public static final int DVB_J_application
```

The DVB registered value for all DVB-J applications.

Methods

getAppIcon()

```
public org.dvb.application.AppIcon getAppIcon()
```

This method returns an object encapsulating the information about the icon(s) for the application.

Returns:

the information related to the icons that are attached to the application or null if no icon information is available.

Since:

MHP1.0.

getIdentifier()

```
public org.dvb.application.AppID getIdentifier()
```

This method returns the application identifier.

Returns:

the application identifier.

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

Since:

MHP1.0.

getIsServiceBound()

```
public boolean getIsServiceBound()
```

This method determines whether the application is bound to a single service.

Returns:

true if the application is bound to a single service, false otherwise.

Since:

MHP1.0.

getName()

```
public java.lang.String getName()
```

This method returns the name of the application. If the default language (as specified in user preferences) is in the set of available language / name pairs then the name in that language shall be returned. Otherwise this method will return a name which appears in that set on a "best-effort basis". If no application names are signalled, an empty string shall be returned.

Returns:

the name of the application.

Since:

MHP1.0.

getName(String)

```
public java.lang.String getName(java.lang.String iso639code)
throws LanguageNotAvailableException
```

This method returns the name of the application in the language which is specified by the parameter passed as an argument. If the language specified is not in the set of available language /name pairs then an exception shall be thrown.

Parameters:

`iso639code` - the specified language, encoded as per ISO 639 [84].

Returns:

returns the name of the application in the specified language.

Throws:

`LanguageNotAvailableException` - if the name is not available in the language specified or if the parameter passed is null.

Since:

MHP1.0.

getNames()

```
public java.lang.String[][] getNames()
```

This method returns all the available names for the application together with their ISO 639 [84] language code. If no application names are signalled, an array of length zero shall be returned.

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

Returns:

the possible names of the application, along with their ISO 639 [84] language code. The first string in each sub-array is the ISO 639 [84] language code. The second string in each sub-array is the corresponding application name.

Since:

MHP1.0.

getPriority()

```
public int getPriority()
```

This method returns the priority of the application.

Returns:

the priority of the application.

Since:

MHP1.0.

getProfiles()

```
public java.lang.String[] getProfiles()
```

This method returns those minimum profiles required for the application to execute. Profile names shall be encoded using the same encoding specified elsewhere in this specification as input for use with the `java.lang.System.get-Property` method to query if a profile is supported by this platform.

For example, for implementations conforming to the first version of the specification, the translation from AIT signaling values to strings shall be as follows:

- '1' in the signaling will be translated into 'mhp.profile.enhanced_broadcast';
- '2' in the signaling will be translated into 'mhp.profile.interactive_broadcast'.

Only profiles supported by this particular GEM terminal shall be returned. Hence the method can return an array of size zero where all the profiles on which an application can execute are unknown.

Returns:

an array of Strings, each String describing a profile.

Since:

MHP1.0.

getProperty(String)

```
public java.lang.Object getProperty(java.lang.String index)
```

The following method is included for properties that do not have explicit property accessors. The naming of properties and their return values are described in the main body of the present document.

Parameters:

`index` - a property name.

Returns:

either the return value corresponding to the property name or null if the property name is unknown or null.

Since:

MHP1.0.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 558 of 861

getServiceLocator()

```
public org.davic.net.Locator getServiceLocator()
```

This method returns the locator of the Service describing the application. For an application transmitted on a remote connection, the returned locator shall be the service for that remote connection. For applications not transmitted on a remote connection, the service returned shall be the currently selected service of the service context within which the application calling the method is running.

Returns:

the locator of the Service describing the application.

Since:

MHP1.0.

getType()

```
public int getType()
```

This method returns the type of the application (as registered by DVB).

Returns:

the type of the application (as registered by DVB).

Since:

MHP1.0.

getVersions(String)

```
public int[] getVersions(java.lang.String profile)
throws IllegalProfileParameterException
```

This method returns an array of integers containing the version number of the specification required to run this application at the specified profile.

Parameters:

`profile` - a profile encoded as defined in the clause "Profile and version properties" in the main body of the present document. e.g. mhp.profile.interactive_broadcast for the interactive broadcast profile.

Returns:

an array of integers, containing the major, minor and micro values (in that order) required for the specified profile.

Throws:

`IllegalProfileParameterException` - thrown if the profile specified is not one of the minimum profiles required for the application to execute or if the parameter passed in is null.

Since:

MHP1.0.

isStartable()

```
public boolean isStartable()
```

This method determines whether the application is startable or not. An Application is not startable if any of the following apply.

- The application is transmitted on a remote connection, and either does not have an application storage descriptor, is not cached, or is not signalled as launchable completely from cache.

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

- The application is signalled as not_launchable_from_broadcast and is not stored at the time the AppAttributes instance is first returned to an application.
- The caller of the method does not have the Permissions to start it.
- if the application is signalled with a control code which is neither AUTOSTART nor PRESENT.

If none of the above apply, then the application is startable.

The value returned by this method does not depend on whether the application is actually running or not.

Returns:

true if an application is startable, false otherwise.

Since:

MHP1.0.

isVisible()

```
public boolean isVisible()
```

This method determines whether the application is marked as being visible to users. An inter-operable application shall honour this visibility setting. Thus a generic launching application shall list applications that are marked as visible and shall not list applications that are not marked as visible.

Returns:

`true` if this application is marked as being visible to users, `false` otherwise.

Since:

MHP1.0.3.

org.dvb.application

AppIcon

Declaration

```
public class AppIcon

java.lang.Object
  |
  +--org.dvb.application.AppIcon
```

Description

The `AppIcon` encapsulates the information concerning the icon attached to the application.

Constructors

AppIcon()

```
protected AppIcon()
```

The constructor for the class. This constructor is intended for implementation convenience and evolution of the specification and not for use by GEM applications. Applications should obtain instances of this class from `AppAttributes.getAppIcon`.

See Also:

```
AppAttributes.getAppIcon()
```


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

Methods

getIconFlags()

```
public java.util.BitSet getIconFlags()
```

This method returns the flags identifying which icons are provided for the application.

Returns:

the icon flags encoded as a BitSet.

Since:

MHP1.0.

getLocator()

```
public org.davic.net.Locator getLocator()
```

This method returns the location of the directory containing the application icons.

Returns:

the location of the directory containing the application icons.

Since:

MHP1.0.

org.dvb.application

AppID

Declaration

```
public class AppID

java.lang.Object
  |
  +--org.dvb.application.AppID
```

Description

The `AppID` is a representation of the unique identifier for applications.

Its string form is the Hex representation of the 48 bit number.

Constructors

AppID(int, int)

```
public AppID(int oid, int aid)
```

Create a new AppID based on the given integers. There is no range checking on these numbers.

Parameters:

`oid` - the globally unique organization number.

`aid` - the unique count within the organization.

Since:

MHP1.0.

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

Methods

equals(Object)

```
public boolean equals(java.lang.Object obj)
```

Compares two AppIDs for equality.

Overrides:

`equals` in class `Object`

Parameters:

`obj` - the reference object with which to compare.

Returns:

`true` if this obj is an AppID and its Organisation ID and its Application ID match the IDs for this AppID; `false` otherwise.

getAID()

```
public int getAID()
```

This method returns the integer value of the application count supplied in the constructor.

Returns:

the integer value of the application count supplied in the constructor.

Since:

MHP1.0.

getOID()

```
public int getOID()
```

This method returns the integer value of the organization number supplied in the constructor.

Returns:

the integer value of the organization number supplied in the constructor.

Since:

MHP1.0.

hashCode()

```
public int hashCode()
```

Returns a hash code value for this AppID. The hashcode for two AppIDs with the same Organisation ID and Application ID are equal.

Overrides:

`hashCode` in class `Object`

Returns:

a hash code value for this AppID.

toString()

```
public java.lang.String toString()
```


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

This method returns a string containing the Hex representation of the 48 bit number. The string shall be formatted as specified in the clause on "Text encoding of application identifiers" in the System Integration clause of the GEM specification.

Overrides:

`toString` in class `Object`

Returns:

a string containing the Hex representation of the 48 bit number.

Since:

MHP1.0.

org.dvb.application

AppProxy

Declaration

```
public interface AppProxy
```

All Known Subinterfaces:

`DVBJProxy`

Description

An `AppProxy` Object is a proxy to an application. A call to the start, stop or pause will cause the resident Application Manager to respectively start, stop or pause the application bound to this AppProxy object. Each of these three method calls can throw a Security Exception if the calling application is not entitled to do so.

Each of these method calls are asynchronous and will result in exactly one AppStateChangeEvent to be generated whether the method call was successful or not. If the method call was not successful, any call to the `hasFailed` method of the corresponding `AppStateChangeEvent` will return true.

Some of the methods here allow the AppProxy to transition through several states before the final state is reached. If this compound state transition is unsuccessful at any point, the resulting AppStateChangeEvent shall have a fromstate which is the last state in this transition which the AppProxy successfully entered and a toState which would have been the next state in the compound state transition.

For instance, if an application were to call start on an AppProxy for a DVB-J application in the NOT_LOADED state and that DVB-J application was to throw a XletStateChangeException from its startXlet method, the getFromState will return PAUSED and getToState will return STARTED. If an application were to call start on an AppProxy for a DVB-J application in the NOT_LOADED state and that DVB-J application was to throw a XletStateChangeException from its initXlet method, the getFromState will return NOT_LOADED and getToState will return PAUSED. Calling the start method for an application which is already running shall fail and generate an AppStateChangeEvent with hasFailed returning true and both fromstate and tostate being STARTED.

See the definition of `AppStateChangeEvent` for more information. Having more than one application attempt to control the lifecycle of the same (other) application at the same time is not recommended. The ordering of AppState-ChangeEvents if this is done will be implementation dependent.

See Also:

`AppStateChangeEvent`

Fields

DESTROYED

```
public static final int DESTROYED
```


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

The application is in the destroyed state. 1 The application instance is in the destroyed state and remains in this state. An AppProxy for a new instance of this application may be obtained from the AppsDatabase and started.

INVALID

```
public static final int INVALID
```

The application is not signalled in the currently selected service and hence not listed in the applications database. Once an AppProxy is in this state, all attempts to cause further state transitions on that AppProxy instance shall fail with an AppStateChangeEvent from INVALID to INVALID, even if a service is selected where that application is signalled.

NOT_LOADED

```
public static final int NOT_LOADED
```

The application has not yet been loaded from the network at all.

PAUSED

```
public static final int PAUSED
```

The application is in the paused state.

STARTED

```
public static final int STARTED
```

The application is in the active state.

Methods

addAppStateChangeEventListener(AppStateChangeEventListener)

```
public void addAppStateChangeEventListener(org.dvb.application.AppStateChangeEventListener listener)
```

Add a listener to the application proxy so that an application can be informed if the application changes state.

Parameters:

`listener` - the listener to be added.

Since:

MHP1.0.

getState()

```
public int getState()
```

Return the current state of the application.

Returns:

the state of the application.

pause()

```
public void pause()
```

Request that the application manager pause the application bound to this information structure.

The `application` will be paused. Calls to this method shall fail if the application is not in the active state. If the application represented by this AppProxy is a DVB-J application, calling this method will, if successful, result in the `pauseXlet` method being called on the Xlet making up the DVB-J application.

Throws:

`java.lang.SecurityException` - if the application is not entitled to pause this application. Note that if an application is entitled to stop an application, it is also entitled to pause it: having the right to stop an application is logically equivalent to having the right to pause it.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 564 of 861

Since:

MHP1.0.

removeAppStateChangeEventListener(AppStateChangeEventListener)

```
public void removeAppStateChangeEventListener(org.dvb.application.AppStateChangeEventListener
listener)
```

Remove a listener on the database.

Parameters:

`listener` - the listener to be removed.

Since:

MHP1.0.

resume()

```
public void resume()
```

Request that the application manager resume the execution of the application. The `application` will be started. This method will throw a security exception if the application does not have the authority to resume the application. Calls to this method shall fail if the application is not in the paused state.

This method is asynchronous and its completion will be notified by an AppStateChangeEvent. In case of failure, the hasFailed method of the `AppStateChangeEvent` will return true. If the application represented by this AppProxy is a DVB-J application, calling this method will, if successful, result in the `startXlet` method being called on the Xlet making up the DVB-J application.

Throws:

`java.lang.SecurityException` - if the application is not entitled to resume this application.

Since:

MHP1.0.

start()

```
public void start()
```

Request that the application manager start the application bound to this information structure.

The `application` will be started. This method will throw a security exception if the application does not have the authority to start applications. Calls to this method shall only succeed if the application if the application is signalled with a control code which is either AUTOSTART or PRESENT and any one of the following applies:

- if the application (DVB-J or DVB-HTML) is in the not loaded or paused states;
- if a DVB-J application is in the "loaded" state;
- if a DVB-HTML application is in the "loading" state.

If the application was not loaded at the moment of this call, then the application will be started. In the case of a DVB-J application, it will be initialized and then started by the Application Manager, hence causing the Xlet to go from Not-Loaded to Paused and then from Paused to Active. If the application was in the Paused state at the moment of the call and had never been in the Active state, then the application will be started. If the application represented by this App-Proxy is a DVB-J application, calling this method will, if successful, result in the `startXlet` method being called on the Xlet making up the DVB-J application.

This method is asynchronous and its completion will be notified by an AppStateChangeEvent. In case of failure, the hasFailed method of the `AppStateChangeEvent` will return true.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 565 of 861

Throws:

`java.lang.SecurityException` - if the application is not entitled to start this application.

Since:

MHP1.0.

start(String[])

```
public void start(java.lang.String[] args)
```

Request that the application manager start the application bound to this information structure passing to that application the specified parameters.

The `application` will be started. This method will throw a security exception if the application does not have the authority to start applications. Calls to this method shall only succeed if the application if the application is signalled with a control code which is either AUTOSTART or PRESENT and any one of the following applies:

- if the application (DVB-J or DVB-HTML) is in the not loaded or paused states;
- if a DVB-J application is in the "loaded" state;
- if a DVB-HTML application is in the "loading" state.

If the application was not loaded at the moment of this call, then the application will be started. In the case of a DVB-J application, it will be initialized and then started by the Application Manager, hence causing the Xlet to go from Not-Loaded to Paused and then from Paused to Active. If the application was in the Paused state at the moment of the call and had never been in the Active state, then the application will be started. If the application represented by this App-Proxy is a DVB-J application, calling this method will, if successful, result in the `startXlet` method being called on the Xlet making up the DVB-J application.

This method is asynchronous and its completion will be notified by an AppStateChangeEvent. In case of failure, the hasFailed method of the `AppStateChangeEvent` will return true. If this AppProxy is a DVBJProxy and if init(String[]) has previously been called for this application instance, then the parameters are not passed into the application but are silently discarded. Those passed into init take precedence.

Parameters:

`args` - the parameters to be passed into the application being started.

Throws:

`java.lang.SecurityException` - if the application is not entitled to start this application.

Since:

MHP1.0.1.

stop(boolean)

```
public void stop(boolean forced)
```

Request that the application manager stop the application bound to this information structure.

The `application` will be stopped. A call to this method shall fail if the application was already in the destroyed state. This method call will stop the application if it was in any other state before the call. If the application is in the NOT_LOADED state then it shall move directly to the DESTROYED state with no other action being taken. If the application represented by this AppProxy is a DVB-J application and is not in the DESTROYED state then calling this method will, if successful, result in the `destroyXlet` method being called on the Xlet making up the DVB-J application with the same value for the parameter as passed to this method.

This method is asynchronous and its completion will be notified by an AppStateChangeEvent. In case of failure, the hasFailed method of the `AppStateChangeEvent` will return true.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 566 of 861

Parameters:

`forced` - if true then do not ask the application but forcibly terminate it, if false give the application an opportunity to refuse.

Throws:

`java.lang.SecurityException` - if the application is not entitled to stop this application.

Since:

MHP1.0.

org.dvb.application

AppsControlPermission

Declaration

```
public final class AppsControlPermission extends java.security.BasicPermission

java.lang.Object
  |
  +--java.security.Permission
        |
        +--java.security.BasicPermission
              |
              +--org.dvb.application.AppsControlPermission
```

All Implemented Interfaces:

`java.security.Guard, java.io.Serializable`

Description

This class represents a Permission to control the lifecycle of another application.

Constructors

AppsControlPermission()

```
public AppsControlPermission()
```

Creates a new AppsControlPermission. There is a simple mapping between the Application control Permission requests and the way the AppsControlPermission are granted. This mapping is defined in the main body of the present document.

AppsControlPermission(String, String)

```
public AppsControlPermission(java.lang.String name, java.lang.String actions)
```

Creates a new AppsControlPermission. There is a simple mapping between the Application control Permission requests and the way the AppsControlPermission are granted. This mapping is defined in the main body of the present document. The actions string is currently unused and should be null. The name string is currently unused and should be empty. This constructor exists for use by the java.security.Policy object to instantiate new permission objects.

Parameters:

`name` - the name of the permission.

`actions` - the actions string.

Methods

equals(Object)

```
public boolean equals(java.lang.Object obj)
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 567 of 861

Checks for equality against this AppsControlPermission object.

Overrides:

`equals` in class `BasicPermission`

Parameters:

`obj` - the object to test for equality with this AppsControlPermission object.

Returns:

true if and only if obj is an AppsControlPermission.

getActions()

```
public java.lang.String getActions()
```

Returns the list of actions that had been passed to the constructor - it shall return null.

Overrides:

`getActions` in class `BasicPermission`

Returns:

a null String.

hashCode()

```
public int hashCode()
```

Returns the hash code value for this object.

Overrides:

`hashCode` in class `BasicPermission`

Returns:

the hash code value for this object.

implies(Permission)

```
public boolean implies(java.security.Permission permission)
```

Checks if this AppsControlPermission object "implies" the specified permission.

Overrides:

`implies` in class `BasicPermission`

Parameters:

`permission` - the specified permission to check.

Returns:

true if and only if the specified permission is an instanceof AppsControlPermission.

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

org.dvb.application

AppsDatabase

Declaration

```
public class AppsDatabase

java.lang.Object
  |
  +--org.dvb.application.AppsDatabase
```

Description

The `AppsDatabase` is an abstract view of the currently available applications. The entries will be provided by the application manager, and gleaned from the AIT signaling. When the service context in which an application is running undergoes service selection, instances of `AppsDatabase` used by that application shall be updated from the new service before an `AppsDatabaseEvent` is sent to the `newDatabase` method of any registered `AppsDatabaseEventListeners`. For applications fully signalled in the current service (i.e. excluding externally authorised ones), the attributes entries shall be the ones from the signalling of the current service even if the application was originally launched from another service and then survived service selection. For running externally authorised applications, the entries will be those from the last service in which they ran fully signalled.

Externally authorized applications shall not appear unless an instance of that application is actually running.

A generic launcher may be written which uses the database to display information in AppAttributes and uses an App-Proxy to launch it

Methods on classes in this package do not block, they return the information the system currently has. Therefore applications should be aware that data may be stale, to within one refresh period of the AIT.

e.g.:

```
AppsDatabase theDatabase = AppsDatabase.getAppsDatabase();
if (theDatabase != null ) {
    CurrentServiceFilter filter = new CurrentServiceFilter();
    Enumeration attributes = theDatabase.getAppAttributes(filter)
    if(attributes != null) {
       while(attributes.hasMoreElements()) {
           AppAttributes info ;
           AppProxy proxy ;

           info = (AppAttributes)attributes.nextElement();
           proxy = (AppProxy)theDatabase.getAppProxy(info.getIdentifier());
           AppIcon icon = info.getAppIcon();
           // blah blah..
           // lets start it.
           proxy.start();
       }
    }
}
```

Where methods on this class as specified as working on "available" applications or "currently available" applications the following definition shall apply. An application is "currently available" if and only if one of the following applies in the service context within which the application calling the method is executing and the visibility of the application is not '00'.

- It is signalled as being present or autostart in the currently selected service of that service context and could be started.
- It is signalled as being present or autostart in the currently selected service of that service context, it could be started if it was stored or cached on the terminal, it is signalled such that it could be stored or cached, and the GEM terminal supports application storage (i.e. the `mhp.stored.services` property is `"SUPPORTED"`). (Note this is **not** affected by whether or not the GEM terminal has sufficient storage space to store the application in question).

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

- It is currently running in that service context.

In addition to the methods listed below, all calls made using an AppsDatabaseFilter shall only use that filter to test "currently available" applications as defined here.

Applications whose information (e.g. signaling) is invalid (e.g. one or more mandatory descriptors are missing or incorrect) may not be listed in the AppsDatabase. Where applications are signalled in a broadcast AIT and the GEM terminal tunes away from the service on which the AIT is carried, but without selecting a new service, the `AppsDatabase` shall retain the entries as signalled in that AIT until a new service is selected.

Constructors

AppsDatabase()

```
protected AppsDatabase()
```

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

addListener(AppsDatabaseEventListener)

```
public void addListener(org.dvb.application.AppsDatabaseEventListener listener)
```

Add a listener to the database so that an application can be informed if the database changes.

Parameters:

`listener` - the listener to be added.

Since:

MHP1.0.

getAppAttributes(AppID)

```
public org.dvb.application.AppAttributes getAppAttributes(org.dvb.application.AppID key)
```

Returns the properties associated with the given ID. Returns null if no such application is available.

Only one AppAttributes object shall be returned in the case where there are several applications having the same (organisationId, applicationId) pair. In such a case, the same algorithm as would be used to autostart such applications shall be used to decide between the available choices by the implementation.

This method shall return instances which reflect the contents of the database at the time the method is called. After an AppsDatabaseEvent has been generated, new instances may be returned. After a service selection has taken place, applications which survived the service selection may call this method in order to discover the attributes of the applications signalled on the new service.

Parameters:

`key` - an application ID.

Returns:

the value to which the key is mapped in this dictionary if AppId corresponds to an application which is either a currently available application or remote application or both. Null otherwise.

Since:

MHP1.0.

getAppAttributes(AppsDatabaseFilter)

```
public java.util.Enumeration getAppAttributes(org.dvb.application.AppsDatabaseFilter filter)
```


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

Returns an enumeration of AppAttributes of the applications available. The Enumeration will contain the set of App-Attributes that satisfy the filtering criteria.

This method shall return instances which reflect the contents of the database at the time the method is called. After an AppsDatabaseEvent has been generated, new instances may be returned. After a service selection has taken place, applications which survived the service selection may call this method in order to discover the attributes of the applications signalled on the new service.

This method will return an empty Enumeration if there are no attributes.

Parameters:

`filter` - the filter to apply.

Returns:

an enumeration of the applications attributes.

Since:

MHP1.0.

getAppIDs(AppsDatabaseFilter)

```
public java.util.Enumeration getAppIDs(org.dvb.application.AppsDatabaseFilter filter)
```

Returns an enumeration of the application IDs available. The Enumeration will contain the set of AppID that match the filtering criteria.

This method shall return instances which reflect the contents of the database at the time the method is called. After an AppsDatabaseEvent has been generated, new instances may be returned. After a service selection has taken place, applications which survived the service selection may call this method in order to discover the identities of the applications signalled on the new service.

This method will return an empty Enumeration if there are no matching applications.

Parameters:

`filter` - the filter to apply.

Returns:

the applications available matching the filtering criteria.

Since:

MHP1.0.

getAppProxy(AppID)

```
public org.dvb.application.AppProxy getAppProxy(org.dvb.application.AppID key)
```

Returns the ApplicationProxy associated with the given ID. Returns null if no such application available.

Only one AppProxy object shall be returned in the case where there are several applications having the same (organisationId, applicationId) pair. In such a case, the same algorithm as would be used to autostart such applications shall be used to decide between the available choices by the implementation.

If an application has an application instance in the destroyed state then a proxy for that appplication instance shall not be retrieved. Instead, what shall be retrieved is a proxy for another application instance which shall be in the not loaded state unless that application instance has already been started.

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

Parameters:

`key` - an application ID.

Returns:

the AppProxy associated with the key parameter or null if the key is not an application ID, or not mapped to any application available.

Throws:

`java.lang.SecurityException` - shall not be thrown for AppIDs which are returned by getAppIDs(CurrentServiceFilter) or getAppIDs(RunningApplicationsFilter).

Since:

MHP1.0.

getAppsDatabase()

```
public static org.dvb.application.AppsDatabase getAppsDatabase()
```

Returns the singleton AppsDatabase object. The AppsDatabase is either a singleton for each GEM application or a singleton for the GEM terminal.

Returns:

the singleton AppsDatabase object.

Since:

MHP1.0.

removeListener(AppsDatabaseEventListener)

```
public void removeListener(org.dvb.application.AppsDatabaseEventListener listener)
```

Remove a listener on the database.

Parameters:

`listener` - the listener to be removed.

Since:

MHP1.0.

size()

```
public int size()
```

Returns the number of applications currently available.

Returns:

the number of applications currently available.

Since:

.HP1.0

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

org.dvb.application

AppsDatabaseEvent

Declaration

```
public class AppsDatabaseEvent extends java.util.EventObject

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.application.AppsDatabaseEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

The `AppsDatabaseEvent` class indicates either an entry in the application database has changed, or so many changes have occurred. that the database should be considered totally new. An event with event_id NEW_DATABASE shall always be sent after switching to a new service. After such an event, the contents of the database (both the set of applications and their attributes) shall reflect the new database contents. All former contents of the database shall be discarded except for running externally authorised applications. It is platform dependant if and when a new database event is thrown while tuned to the same service except that a NEW_DATABASE event shall not be sent when only one application has changed within a service.

The APP_ADDED, APP_CHANGED and APP_DELETED events shall not be generated in response to the same database change as caused a NEW_DATABASE event to be generated.

Since:

MHP1.0.

Fields

APP_ADDED

```
public static final int APP_ADDED
```

The addition event id. The APP_ADDED event is generated whenever an entry is added to the AppsDatabase. It is NOT generated when the entry already in the AppsDatabase changes.

APP_CHANGED

```
public static final int APP_CHANGED
```

The changed event id. The APP_CHANGED event is generated whenever any of the information about an application changes. It is NOT generated when the entry is added to or removed from the AppsDatabase. In such cases, the APP_ADDED or APP_DELETED events will be generated instead.

APP_DELETED

```
public static final int APP_DELETED
```

The deletion event id. The APP_DELETED event is generated whenever an entry is removed from the AppsDatabase.

NEW_DATABASE

```
public static final int NEW_DATABASE
```

The new database event id.

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

Constructors

AppsDatabaseEvent(int, AppID, Object)

```
public AppsDatabaseEvent(int id, org.dvb.application.AppID appid, java.lang.Object source)
```

Create a new AppsDatabaseEvent object for the entry in the database that changed, or for a new database.

Parameters:

`id` - the cause of the event.

`appid` - the AppId of the entry that changed.

`source` - the AppaDatabase object.

Since:

MHP1.0.

Methods

getAppID()

```
public org.dvb.application.AppID getAppID()
```

gets the application ID object for the entry in the database that changed.

When the event type is NEW_DATABASE, AppID will be null.

Returns:

application ID representing the application.

Since:

MHP1.0.

getEventId()

```
public int getEventId()
```

gets the type of the event.

Returns:

an integer that matches one of the static fields describing events.

Since:

MHP1.0.

org.dvb.application

AppsDatabaseEventListener

Declaration

```
public interface AppsDatabaseEventListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```


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

Description

The `AppsDatabaseListener` class allows an application to monitor the application database so that it can keep an up to date interface without polling the state. The application shall receive these events in a timely fashion after the AIT changes, however it is system dependant how often the AIT table is checked.

Since:

MHP1.0.

Methods

entryAdded(AppsDatabaseEvent)

```
public void entryAdded(org.dvb.application.AppsDatabaseEvent evt)
```

The AppsDataBase has had an application entry added.

Parameters:

`evt` - the AppsDatabaseEvent.

Since:

MHP1.0.

entryChanged(AppsDatabaseEvent)

```
public void entryChanged(org.dvb.application.AppsDatabaseEvent evt)
```

The AppsDataBase has had an application entry changed.

Parameters:

`evt` - the AppsDatabaseEvent.

Since:

MHP1.0.

entryRemoved(AppsDatabaseEvent)

```
public void entryRemoved(org.dvb.application.AppsDatabaseEvent evt)
```

The AppsDataBase has had an application entry removed.

Parameters:

`evt` - the AppsDatabaseEvent.

Since:

MHP1.0.

newDatabase(AppsDatabaseEvent)

```
public void newDatabase(org.dvb.application.AppsDatabaseEvent evt)
```

The AppsDataBase has radically changed.

Parameters:

`evt` - the AppsDatabaseEvent.

Since:

MHP1.0.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 575 of 861

org.dvb.application

AppsDatabaseFilter

Declaration

```
public abstract class AppsDatabaseFilter

java.lang.Object
  |
  +--org.dvb.application.AppsDatabaseFilter
```

Direct Known Subclasses:

```
CurrentServiceFilter, RunningApplicationsFilter
```

Description

Abstract class for the filters. Instances of concrete classes that extend `AppsDatabaseFilter` are passed to the AppsDatabase.getAppAttributes and AppsDatabase.getAppIDs methods to allow an applications to set a filter on the list of applications (respectively AppAttributes and AppIDs) that it wants to retrieve from the AppDatabase.

Since:

MHP 1.0.

Constructors

AppsDatabaseFilter()

```
public AppsDatabaseFilter()
```

Construct an AppsDatabaseFilter object.

Methods

accept(AppID)

```
public abstract boolean accept(org.dvb.application.AppID appid)
```

Test if a specified appid should be included in the Enumeration.

Parameters:

`appid` - the specified appid to test.

Returns:

true if the application with identifier appid should be listed, false otherwise.

Since:

MHP1.0.

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

org.dvb.application
AppStateChangeEvent

Declaration

```
public class AppStateChangeEvent extends java.util.EventObject

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.application.AppStateChangeEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

The `AppStateChangeEvent` class indicates a state transition of the application. These events are only generated for running applications or for non-running applications where an attempt to control the application fails. If the state transition was requested by an application through this API, the method `hasFailed` indicates whether the state change failed or not. Where a state change succeeds, `fromState` and `toState` shall indicate the original and destination state of the transition. If it failed, `fromState` shall return the state the application was in before the state transition was requested and the `toState` method shall return the state the application would have been in if the state transition had succeeded.

Attempting to start an application which is already in the active state shall fail and generate an `AppStateChange-Event` with `hasFailed` returning true and both fromstate and tostate being `STARTED`.

Since:

MHP1.0.

Constructors

AppStateChangeEvent(AppID, int, int, Object, boolean)

```
public AppStateChangeEvent(org.dvb.application.AppID appid, int fromstate, int tostate,
java.lang.Object source, boolean hasFailed)
```

Create an AppStateChangeEvent object.

Parameters:

`appid` - a registry entry representing the tracked application.

`fromstate` - the state the application was in before the state transition was requested, where the value of fromState is one of the state values defined in the AppProxy interface or in the interfaces inheriting from it.

`tostate` - state the application would be in if the state transition succeeds, where the value of toState is one of the state values defined in the AppProxy interface or in the interfaces inheriting from it.

`hasFailed` - an indication of whether the transition failed (true) or succeeded (false).

`source` - the `AppProxy` where the state transition happened.

Methods

getAppID()

```
public org.dvb.application.AppID getAppID()
```

The application the listener was tracking has made a state transition from `fromState` to `toState`.

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

Returns:

a registry entry representing the tracked application.

Since:

MHP1.0.

getFromState()

```
public int getFromState()
```

The application the listener is tracking was in `fromState`, where the value of fromState is one of the state values defined in the AppProxy interface or in the interfaces inheriting from it.

Returns:

the old state.

Since:

MHP1.0.

getToState()

```
public int getToState()
```

If the `hasFailed` method returns false, then the application the listener is tracking is now in `toState`. If the `hasFailed` method returns true, then the `toState` is the state where the state transition was attempted to but the transition failed. The value of `toState` is one of the state values defined in the `AppProxy` interface or in the interfaces inheriting from it.

Returns:

the intended or actual new state.

Since:

MHP1.0.

hasFailed()

```
public boolean hasFailed()
```

This method determines whether an attempt to change the state of an application has failed.

Returns:

true if the attempt to change the state of the application failed, false otherwise.

Since:

MHP1.0.

org.dvb.application

AppStateChangeEventListener

Declaration

```
public interface AppStateChangeEventListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```


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

Description

The `AppStateChangeEventListener` class allows a launcher application to keep track of applications it launches or other applications running as part of the same service.

Since:

MHP1.0.

Methods

stateChange(AppStateChangeEvent)

```
public void stateChange(org.dvb.application.AppStateChangeEvent evt)
```

The application the listener was tracking has made a state transition from `fromState` to `toState` and this method will be given the state event.

Parameters:

`evt` - the AppStateChangeEvent.

Since:

MHP1.0.

org.dvb.application

CurrentServiceFilter

Declaration

```
public class CurrentServiceFilter extends AppsDatabaseFilter
```

```
java.lang.Object
  |
  +--org.dvb.application.AppsDatabaseFilter
        |
        +--org.dvb.application.CurrentServiceFilter
```

Description

Instances of `CurrentServiceFilter` are used to set a filter on the list of applications that are retrieved from the Apps-Database (See methods getAppsAttributes and getAppsIDs).

A `CurrentServiceFilter` is used to indicate that only applications that signalled as part of the current service shall be returned by the getAppsAttributes and getAppIDs methods of AppsDatabase. Externally authorized applications in the AIT are not considered to be signalled as part of the current service for this filter. Subclasses of CurrentServiceFilter can override the accept method so as to implement their own filter criteria on the AppIDs values.

Since:

MHP 1.0.

Constructors

CurrentServiceFilter()

```
public CurrentServiceFilter()
```

public Constructor of the CurrentServiceFilter.

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

Methods

accept(AppID)

```
public boolean accept(org.dvb.application.AppID appid)
```

Test if a specified appid should be included in the Enumeration.

Overrides:

`accept` in `class AppsDatabaseFilter`

Parameters:

`appid` - the specified appid to test.

Returns:

true if the application with identifier appid should be listed, false otherwise.

Since:

MHP1.0.

org.dvb.application

DVBJProxy

Declaration

```
public interface DVBJProxy extends AppProxy
```

All Superinterfaces:

`AppProxy`

Description

A `DVBJProxy` Object is a proxy to a DVBJ application.

Fields

LOADED

```
public static final int LOADED
```

The application is in the loaded state.

Methods

init()

```
public void init()
```

Requests the application manager calls the `initXlet` method on the application.

This method is asynchronous and its completion will be notified by an AppStateChangeEvent. In case of failure, the hasFailed method of the `AppStateChangeEvent` will return true. Calls to this method shall only succeed if the application is in the NOT_LOADED or LOADED states. If the application is in the NOT_LOADED state, the application will move through the LOADED state into the PAUSED state before calls to this method complete.

In all cases, an AppStateChangeEvent will be sent, whether the call was successful or not.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 580 of 861

Throws:

`java.lang.SecurityException` - if the application is not entitled to load this application. Being able to init an application requires to be entitled to start it.

Since:

MHP1.0.

init(String[])

```
public void init(java.lang.String[] args)
```

Requests the application manager calls the `initXlet` method on the application passing to that application the specified parameters.

This method is asynchronous and its completion will be notified by an AppStateChangeEvent. In case of failure, the hasFailed method of the `AppStateChangeEvent` will return true. Calls to this method shall only succeed if the application is in the NOT_LOADED or LOADED states. If the application is in the NOT_LOADED state, the application will move through the LOADED state into the PAUSED state before calls to this method complete.

In all cases, an AppStateChangeEvent will be sent, whether the call was successful or not.

Parameters:

`args` - the parameters to be passed into the application being started.

Throws:

`java.lang.SecurityException` - if the application is not entitled to load this application. Being able to init an application requires to be entitled to start it.

Since:

MHP1.1.2.

load()

```
public void load()
```

Provides a hint to preload at least the initial class of the application into local storage, resources permitting. This does not require loading of classes into the virtual machine or creation of a new logical virtual machine which are implications of the `init` method.

This method is asynchronous and its completion will be notified by an `AppStateChangeEvent`. In case of failure, the `hasFailed` method of the `AppStateChangeEvent` will return true. Calls to this method shall only succeed if the application is in the NOT_LOADED state. In all cases, an AppStateChangeEvent will be sent, whether the call was successful or not.

Throws:

`java.lang.SecurityException` - if the application is not entitled to load this application. Being able to load an application requires to be entitled to start it.

Since:

MHP1.0.

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

org.dvb.application IllegalProfileParameterException

Declaration

```
public class IllegalProfileParameterException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.application.IllegalProfileParameterException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

The `IllegalProfileParameter` exception is thrown if the application attempts to ask for a version number for a profile not specified for the application.

Since:

MHP1.0.

Constructors

IllegalProfileParameterException()

```
public IllegalProfileParameterException()
```

Construct a IllegalProfileParameterException with no detail message

IllegalProfileParameterException(String)

```
public IllegalProfileParameterException(java.lang.String s)
```

Construct a IllegalProfileParameterException with a detail message

Parameters:

`s` - detail message.

org.dvb.application LanguageNotAvailableException

Declaration

```
public class LanguageNotAvailableException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.application.LanguageNotAvailableException
```


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

All Implemented Interfaces:

`java.io.Serializable`

Description

The `LanguageNotAvailableException` exception is thrown if the application asks for the name of an application in a language not signalled in the AIT.

Since:

MHP1.0.

Constructors

LanguageNotAvailableException()

```
public LanguageNotAvailableException()
```

Construct a LanguageNotAvailableException with no detail message

LanguageNotAvailableException(String)

```
public LanguageNotAvailableException(java.lang.String s)
```

Construct a LanguageNotAvailableException with a detail message

Parameters:

`s` - detail message.

org.dvb.application RunningApplicationsFilter

Declaration

```
public class RunningApplicationsFilter extends AppsDatabaseFilter
```

```
java.lang.Object
  |
  +--org.dvb.application.AppsDatabaseFilter
        |
        +--org.dvb.application.RunningApplicationsFilter
```

Description

Instances of `RunningApplicationsFilter` are used to set a filter on the list of applications that are retrieved from the AppsDatabase (See methods getAppsAttributes and getAppsIDs).

A `RunningApplicationsFilter` is used to indicate that only applications that are running as part of the current service shall be returned by the getAppsAttributes and getAppIDs methods of AppsDatabase. Externally authorized applications in the AIT shall be returned if they are currently running in the same service context as the caller. Running applications whose visibility is '00' shall not be returned. Subclasses of RunningApplicationsFilter can override the accept method so as to implement their own filter criteria on the AppIDs values.

Since:

MHP 1.0.

Constructors

RunningApplicationsFilter()

```
public RunningApplicationsFilter()
```

public Constructor of the RunningApplicationsFilter

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

Methods

accept(AppID)

```
public boolean accept(org.dvb.application.AppID appid)
```

Test if a specified appid should be included in the Enumeration.

Overrides:

`accept` in class `AppsDatabaseFilter`

Parameters:

`appid` - the specified appid to test.

Returns:

true if the application with identifier appid should be listed, false otherwise.

Since:

MHP1.0.

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

Annex T (normative): Permissions

NOTE 1: See clause 11.10, "Java permissions".

NOTE 2: As a consequence of clause 11.8.3, "Additional permissions classes," the packages `org.dvb.net.ca` and the class `org.dvb.net.tuning.DvbNetworkInterfaceSIUtil` are not required under certain circumstances.

Package

org.dvb.net.ca

Description

Provides extensions to the conditional access API from DAVIC.

Class Summary	
Classes	
`CAPermission`	This class is for CA permissions.

org.dvb.net.ca

CAPermission

Declaration

```
public class CAPermission extends java.security.BasicPermission
```

```
java.lang.Object
  |
  +--java.security.Permission
        |
        +--java.security.BasicPermission
              |
              +--org.dvb.net.ca.CAPermission
```

All Implemented Interfaces:

```
java.security.Guard, java.io.Serializable
```

Description

This class is for CA permissions. A CAPermission contains a name, but no actions list.

A CAPermission contains a range of CA system ids and a specific permission for that range of CA system ids. Instead of a range of CA system ids, the CAPermission can also refer to a single CA system id.

The name has the following syntax:

CASystemIdRange ":" Permission

where CASystemIdRange = CASystemId ["-" CASystemId] | "*"
and Permission = "MMI" | "buy" | "entitlementInfo" | "messagePassing" | "*"

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

EXAMPLES:

- 0x1200-0x120A:buy (The permission to buy entitlement for all the CA systems with ids between 0x1200 and 0x120A inclusive).
- 0x1201:entitlementInfo (The permission to get entitlement information for the CA system with id 0x1201).
- 0x120d:* (This wildcard expresses all the permissions for the CA system with id 0x120d).

NOTE: The CASystemId is expressed as a hexadecimal value.

The permission "MMI" corresponds with the SecurityException on CAModuleManager.addMMIListener(). The permission "buy" corresponds with the SecurityException on CAModule.buyEntitlement(). The permission "entitlementInfo" corresponds with the SecurityException on CAModule.queryEntitlement() and CAModule.list-Entitlements(). The permission "messagePassing" corresponds with CAModule.openMessageSession(Message-Listener).

Constructors

CAPermission(String)

```
public CAPermission(java.lang.String name)
```

Creates a new CAPermission with the specified name. The name is the symbolic name of the CAPermission.

Parameters:

`name` - the name of the CAPermission.

CAPermission(String, String)

```
public CAPermission(java.lang.String name, java.lang.String actions)
```

Creates a new CAPermission object with the specified name. The name is the symbolic name of the CAPermission, and the actions String is unused and should be null. This constructor exists for use by the Policy object to instantiate new Permission objects.

Parameters:

`name` - the name of the CAPermission.

`actions` - should be null.

Methods

implies(Permission)

```
public boolean implies(java.security.Permission p)
```

Checks if the specified permission is "implied" by this object.

Overrides:

`implies` in class `BasicPermission`

Parameters:

`p` - the permission to check against.

Returns:

true if the passed permission is equal to or implied by this permission, false otherwise.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 586 of 861

Package org.dvb.net.tuning

Description

Provides extensions to the tuning API from DAVIC.

Class Summary	
Classes	
DvbNetworkInterfaceSIUtil	Each SI database is associated with a network interface and vice versa.
TunerPermission	This class is for tuner permissions.

org.dvb.net.tuning DvbNetworkInterfaceSIUtil

Declaration

```
public class DvbNetworkInterfaceSIUtil

java.lang.Object
  |
  +--org.dvb.net.tuning.DvbNetworkInterfaceSIUtil
```

Description

Each SI database is associated with a network interface and vice versa. This class allows the application to query this association.

Since:

MHP 1.0.1.

Methods

getNetworkInterface(SIDatabase)

```
public static org.davic.net.tuning.NetworkInterface getNetworkInterface(org.dvb.si.SIDatabase sd)
```

Gets the network interface for a particular SI database.

Parameters:

sd - the SI database for which the associated network interface will be returned.

Returns:

the associated network interface.

getSIDatabase(NetworkInterface)

```
public static org.dvb.si.SIDatabase getSIDatabase(org.davic.net.tuning.NetworkInterface ni)
```

Gets the SI database for a particular network interface.

Parameters:

ni - the network interface for which the associated SI database will be returned.

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

Returns:

the associated SI database.

org.dvb.net.tuning

TunerPermission

Declaration

```
public class TunerPermission extends java.security.BasicPermission

java.lang.Object
  |
  +--java.security.Permission
        |
        +--java.security.BasicPermission
              |
              +--org.dvb.net.tuning.TunerPermission
```

All Implemented Interfaces:

```
java.security.Guard, java.io.Serializable
```

Description

This class is for tuner permissions. The name and actions list of a TunerPermission contains no name and no actions list. The return value of the inherited getName() method is implementation dependent. If an application has the tuner permission, then it shall not receive a `SecurityException` from those methods in that API defined to throw one. Without such a permission, it shall receive such an exception.

Constructors

TunerPermission(String)

```
public TunerPermission(java.lang.String name)
```

Creates a new TunerPermission. The name string is currently unused and should be empty.

Parameters:

`name` - the name of the TunerPermission.

TunerPermission(String, String)

```
public TunerPermission(java.lang.String name, java.lang.String actions)
```

Creates a new TunerPermission. The name string is currently unused and should be empty. The actions string is currently unused and should be null. This constructor exists for use by the Policy object to instantiate new Permission objects.

Parameters:

`name` - the name of the TunerPermission.

`actions` - the actions list.

Methods

implies(Permission)

```
public boolean implies(java.security.Permission p)
```

Checks if the specified permission is "implied" by this object.

Since name and actions are not used, the only check needed is whether p is also a TunerPermission.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 588 of 861

Overrides:

`implies` in class `BasicPermission`

Parameters:

`p` - the permission to check against.

Returns:

true if the passed permission is equal to or implied by this permission, false otherwise.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 589 of 861

Annex U (normative): Extended graphics APIs

Package

org.dvb.ui

Description

Provides extended graphics functionality.

Class Summary	
Interfaces	
`TestOpacity`	Interface implemented by Components or Containers in order to allow the platform to query whether their paint method is fully opaque.
`TextOverflowListener`	The TextOverflowListener is an interface that an application may implement and register in the DVBTextLayoutManager.
Classes	
`BufferedAnimation`	A `BufferedAnimation` is an AWT component that maintains a queue of one or more image buffers.
`DVBAlphaComposite`	This `DVBAlphaComposite` class implements the basic alpha compositing rules for combining source and destination pixels to achieve blending and transparency effects with graphics, images and video.
`DVBBufferedImage`	The `DVBBufferedImage` subclass describes an `java.awt.Image` with an accessible buffer of image data.
`DVBColor`	A Color class which adds the notion of alpha.
`DVBGraphics`	The `DVBGraphics` class is a adapter class to support alpha compositing in a GEM device.
`DVBTextLayoutManager`	The DVBTextLayoutManager provides a text rendering layout mechanism for the org.havi.ui.HStaticText org.havi.ui.HText and org.havi.ui.HTextButton classes.
`FontFactory`	Provides a mechanism for applications to instantiate fonts that are not built into the system.
Exceptions	
`DVBRasterFormatException`	This exception is thrown for some invalid operations on instances of `DVBBufferedImage`.
`FontFormatException`	Thrown when attempt is made to read a file describing a font when the contents of that file are not valid.
`FontNotAvailableException`	Thrown when attempt is made to instantiate a font that cannot be located.
`UnsupportedDrawingOperationException`	The `UnsupportedDrawingOperationException` class represents an exception that is thrown if an drawing operation is not supported on this platform.

org.dvb.ui

BufferedAnimation

Declaration

```
public class BufferedAnimation extends java.awt.Component

java.lang.Object
  |
  +--java.awt.Component
        |
```


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

+--**org.dvb.ui.BufferedAnimation**

All Implemented Interfaces:

`java.awt.image.ImageObserver, java.io.Serializable`

Description

A `BufferedAnimation` is an AWT component that maintains a queue of one or more image buffers. This permits efficient flicker-free animation by allowing a caller to draw to an off-screen buffer, which the system then copies to the framebuffer in coordination with the video output subsystem. This class also allows an application to request a series of buffers, so that it can get a small number of frames ahead in an animation. This allows an application to be robust in the presence of short delays, e.g. from garbage collection. A relatively small number of buffers is recommended, perhaps three or four. A BufferedAnimation with one buffer provides little or no protection from pauses, but does provide double-buffered animation.

This class can be used for frame-synchronous animation. When animation is in progress, it maintains a count of the frame number in the underlying video output device. This frame number increases monotonically by one for each video frame output. It is not influenced by trick play of any video that might be playing on the same screen. However, the framerate of the `BufferedAnimation` may be determined by the framerate of such video; see `setFramerate(float)` and `getFramerate()` for details.

The implementation shall prevent tearing artifacts whenever possible. The maximum size and animation rate that can be achieved without tearing artifacts may be specified by the system model of a specification that includes this class. If it is necessary to avoid a tearing artifact, an implementation shall delay the copying of an internal buffer to the frame buffer by up to one frame.

The size of this component is set using the normal AWT mechanisms. When the method `setBuffers(Dimension, int, boolean, boolean)` is called, initialization is performed. At this time, the size of the graphics buffers is set.

When one of this component's buffers is copied to the frame buffer, it is done without regard to any AWT components which may overlap with this component. This is like the behavior when an application draws directly to the screen using a graphics object obtained with `java.awt.Component.getGraphics()`.

When the system copies a buffer to the frame buffer for a given frame *f*, it shall select the valid buffer associated with the highest-numbered frame *fb* such that *fb* <= *f*. A buffer is valid if `startFrame(int)` has been called for that buffer, `finishFrame(int)` has been called, and the given buffer has not been re-used for a subsequent frame as a result of another call to `startFrame(int)`.

The animation task proceeds at a high relative CPU priority, and can be considered to execute at a priority greater than Java's `Thread.NORM_PRIORITY`. However, when no new image buffer is ready, the system task must always block until one is. Drawing into the buffer is done within a Java thread, which is subject to the normal scheduling guarantees. In this way, a CPU-bound caller can avoid starving more important activities, such as responding to remote control input. CPU-bound applications may wish to invoke `Thread.yield()` after each frame, however, particularly if the application's animation thread is at the same priority level as other application threads.

A component that is not visible and is in the started state will run, but it will not display any buffers to the screen. It will block in the call to `startFrame()` until the component becomes visible, or until it is too late to draw the requested frame, whichever comes first. Once it is too late, it will, of course, return -1, thus ensuring that the caller doesn't waste time drawing to an internal graphics buffer that wouldnt' be displayed.

For the behavior when this component is destroyed, see `removeNotify()`.

Sample usage:

```
BufferedAnimation anim = new BufferedAnimation();
 ... put anim in a component hierarchy
Dimension d = new Dimension(...);
int numBuffers = 4;
for (;;) {
    try {
        anim.setBuffers(d, 4, false, true);
        break;
    } catch (OutOfMemoryError err) {
        ... try smaller buffers, or fewer of them
    }
 }
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 591 of 861

```
 ... set framerate, if needed
 ... Make anim visible
Graphics2D[] bufs = anim.getBuffersGraphics();
anim.startAnimation();
//  Animate 1000 frames...
try {
    for (int f = 0; f < 1000; f++) {
        int n;
        try {
             n = anim.startFrame(f); // blocks until a buffer is free
        } catch (InterruptedException ex) {
             // someone else called stopAnimation. or removeNotify() was
             // called.  In any case, we're being asked to stop the
             // animation immediately.
             break;
        }
        if (n > -1) {
             try {
                 myDrawFrame(f, bufs[n]);
             } finally {
                 anim.finishFrame(f);
             }
         }
    }
 } finally {
     anim.stopAnimation(false);
 }
```

This class does not specify a return value for `Component.isDoubleBuffered()`. That method reports on a different kind of buffering, related to the `repaint()` call. `BufferedAnimation` objects might or might not be double-buffered, in the repaint-related sense meant by `Component.isDoubleBuffered()`.

NOTE: A future version of this API could potentially allow simultaneous drawing to two frames, by relaxing the synchronization condition on startFrame(int). However, it is unclear if this would yield benefits e.g. on a multi-core system, given memory bandwith limitations and the already existing ability to have parallel threads drawing into one frame or doing other computations.

Since:

MHP 1.1.3.

Fields

FRAME_23_98

```
public static float FRAME_23_98
```

Constant representing a common video framerate, approximately 23.98 frames per second, and equal to `24000f/1001f`.

See Also:

`getFramerate(), setFramerate(float)`

FRAME_24

```
public static float FRAME_24
```

Constant representing a common video framerate, equal to `24f`.

See Also:

`getFramerate(), setFramerate(float)`

FRAME_25

```
public static float FRAME_25
```

Constant representing a common video framerate, equal to `25f`.

See Also:

`getFramerate(), setFramerate(float)`

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 592 of 861

FRAME_29_97

```
public static float FRAME_29_97
```

Constant representing a common video framerate, approximately 29.97 frames per second, and equal to `30000f/1001f`.

See Also:

`getFramerate(), setFramerate(float)`

FRAME_50

```
public static float FRAME_50
```

Constant representing a common video framerate, equal to `50f`.

See Also:

`getFramerate(), setFramerate(float)`

FRAME_59_94

```
public static float FRAME_59_94
```

Constant representing a common video framerate, approximately 59.94 frames per second, and equal to `60000f/1001f`.

See Also:

`getFramerate(), setFramerate(float)`

Constructors

BufferedAnimation()

```
public BufferedAnimation()
```

Create a new `BufferedAnimation` component. The BufferedAnimation functionality may be optional. Applications written to device specifications that do not make this functionality mandatory should be prepared to catch UnsupportedOperationException when invoking this constructor.

Throws:

`java.lang.UnsupportedOperationException` - If this feature is not supported on the device.

Methods

addNotify()

```
public void addNotify()
```

Makes this `Component` displayable by connecting it to a native screen resource. This method is called internally by the toolkit and should not be called directly by programs.

Overrides:

`addNotify` in class `Component`

finishFrame(int)

```
public void finishFrame(int frameNumber)
```

Notify the system that the frame currently being drawn is finished. Drawing the current frame is initiated with `startFrame(int)`. Once `finishFrame(int)` is called, the system can copy that frame to the framebuffer, and the caller can move on to preparing the next frame.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 593 of 861

Parameters:

`frameNumber` - The frame number that is finished. This must match the value passed into startFrame(int).

Throws:

`java.lang.IllegalStateException` - if a startFrame call has not returned successfully for the given frame number (with a return value other than -1), if `finishFrame(int)` has already been called for this frame number since the startFrame call, or if `stopAnimation(boolean)` has been called since the corresponding startFrame call.

See Also:

`startFrame(int)`

getBuffersGraphics()

```
public java.awt.Graphics2D[] getBuffersGraphics()
```

Get the graphics objects for drawing into this component's internal image buffers. The size and number of buffers is determined by the `setBuffers` method.

After the first call, subsequent invocations of this method shall return the identical value (i.e. multiple calls will return values that are == to each other).

Other than the `setBuffers` mechanism, the size of the internal buffers will never change. If the component is resized, the system might scale the resulting animation, but this behavior is not guaranteed by the specification of this class. In all cases, the upper-left hand corner of the image buffer will be displayed in the upper-left hand corner of the component.

The initial contents of the graphics buffers is undefined. Callers may wish to initialize the buffers to a known state, such as fully transparent, before starting an animation. Once an animation is started, drawing into a buffer outside of a `startFrame/finishFrame` pair may produce unpredictable results on the screen.

Returns:

An array of graphics objects that can be used to draw into the internal image buffers.

Throws:

`java.lang.IllegalStateException` - if the `setBuffers()` hasn't been successfully called.

See Also:

`setBuffers(Dimension, int, boolean, boolean)`

getBuffersSize()

```
public java.awt.Dimension getBuffersSize()
```

Get the size of the internal image buffers. If setBuffers has not yet been successfully called, then null is returned.

See Also:

`getBuffersGraphics(), setBuffers(Dimension, int, boolean, boolean)`

getFramerate()

```
public float getFramerate()
```

Get the actual framerate of the screen associated with this component. This class defines a number of common framerates as constants whose name begin with "`FRAME_`".

getMediaTime(Clock, int)

```
public long getMediaTime(javax.media.Clock c, int frameNumber)
```

Get the predicted media time of the given frameNumber for this animation. The predicted media time is calculated from the media time of the frame being presented on the screen, and extrapolating assuming a clock rate of 1.0.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 594 of 861

The return value can be converted into a javax.media.Time value by calling the `javax.media.Time(long)` constructor.

This method is useful for keeping an animation aligned with a video source, even if "trick play" operations cause the media position to change. Note that because the computer generated animation might be a small number of frames "ahead" of the video due to the buffering this class provides, there might be a perceptable "lag" during trick play itself, but once the video returns to normal play mode, the animation would once again be frame-synchronized.

Parameters:

`c` - The clock to calculate media time relative to.

`frameNumber` - the desired frame number of this animation.

Returns:

The media time, in nanoseconds.

Throws:

`java.lang.IllegalStateException` - if this animation is not in the started state.

`javax.media.IncompatibleTimeBaseException` - If this Clock is incompatible with this animation. This will never be thrown if the Clock is associated with video being displayed on the same screen as this `BufferedAnimation` component.

`java.lang.IllegalStateException` - If this clock is not in the started state.

isStarted()

```
public boolean isStarted()
```

Return true if this animation is started. A `BufferedAnimation` can either be in the started or stopped state. A stopped `BufferedAnimation` will only draw to the framebuffer if it is in the process of flushing animation buffers due to a call to `stopAnimation(false)`

Returns:

true if this `BufferedAnimation` is started, false otherwise.

See Also:

`stopAnimation(boolean)`

paint(Graphics)

```
public void paint(java.awt.Graphics g)
```

If this component has an active animation, then this method paints either the last valid image buffer or the next valid image buffer to the given graphics object. If there is no available valid image buffer and a `startFrame/finishFrame` sequence is in progress, this method will block until `finishFrame` is called, thus generating a valid image buffer. If no animation is in progress or no valid frames have yet been generated, then this method does nothing.

Note that in normal operation, this method should be called by the platform very infrequently, if at all. It might be called, for example, due to an "expose event," or due to a call to `Component.print(Graphics)`. Application authors should not request a call to `paint` via the repaint mechanism to animate this component, because this class uses a different model for animation.

Overrides:

`paint` in class `Component`

removeNotify()

```
public void removeNotify()
```

Make this component undisplayable by destroying any native resources, and freeing its image buffers. `stopAnimation(true)` shall be called by the implementation of this method.

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

Overrides:

removeNotify in class Component

setBuffers(Dimension, int, boolean, boolean)

```
public void setBuffers(java.awt.Dimension bufSize, int numBuffers, boolean forceSize,
boolean forceAcceleration)
```

Set the size and number of the internal image buffers. If the system is capable of scaling a BufferedAnimation's buffers to the component's size in real-time, then the internal buffers will be of the requested size, and scaling will occur. If the system is not, then the results will depend on the value of forceSize.

On a system that cannot perform a requested scaling, if forceSize is true, then the buffers' sizes will be set to the requested size regardless. The displayed result will be clipped or will have areas that are unpainted, as needed. In all cases, the upper-left hand corner of the buffers will be painted at to the upper-left hand corner of the component.

On a system that cannot perform scaling, if forceSize is false, the buffers' size will be set to the current size of the component. That is, the requested buffer dimension will be ignored.

The system model of specifications that include this class may specify a set of supported scalings.

Note that the framebuffer itself might be scaled for display on the output device. For example, a specification including this class might include half-resolution mode, e.g. for half-resolution computer graphics over 1080i video.

Graphics Acceleration

Some systems have special faster video memory that gives accelerated graphics performance. The system model of a specification adopting this API may define a minimum amount of such memory. Other hardware architectures, such as "unified memory architecture" platforms, don't have special accelerated memory. On these platforms, all video memory is considered "accelerated", that is, the forceAccelerated parameter has no effect, and does not cause automatic failure.

On platforms with special accelerated video memory, the caller may indicate that all of the graphics buffers must be allocated from this accelerated memory. It does this by setting the forceAcceleration parameter true. This may make an OutOfMemoryError more likely. If forceAcceleration is not true, then the implementation will make a "best-effort" attempt to put the buffers in accelerated memory, but will fall back to normal heap memory, if required.

This method may be called more than once. If it exits with an exception, the state of this object will not be changed. If it returns normally, the new values will override anything set previously.

Parameters:

bufSize - The requested buffer size.

numBuffers - The number of image buffers to allocate.

forceSize - Force sizing the buffers to the requested size, even if this means clipping or having unpainted areas.

forceAcceleration - Force allocation of all buffers in accelerated memory.

Throws:

java.lang.IllegalArgumentException - if d.width or d.height is less than one, or numBuffers is less than one.

java.lang.IllegalStateException - if getBuffersGraphics has been called for this component.

java.lang.IllegalStateException - If this component isn't displayable.

OutOfMemoryException - If there isn't enough memory to allocate the needed buffers.

See Also:

java.awt.Component.isDisplayable()

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 596 of 861

setFramerate(float)

```
public void setFramerate(float rate)
```

Attempt to set the framerate of the screen associated with this component. Other factors, such as video being output to the same device or device limitations, might determine the framerate, thus causing this method to have no effect. The frameright might subsequently be changed, e.g. by video being presented on the screen or by other APIs. Unless other behavior is mandated by the system model of a specification incorporating this class, it is an allowable implementation option for this method to never change the framerate.

The system model of specifications including this class might determine under what conditions the framerate can be set, and what framerates are guaranteed to be supported. This class defines a number of common framerates as constants whose name begin with "`FRAME_`".

Note that an application that wishes to animate at a lower framerate than that of the hardware may do so, by simply skipping frames. This is discussed in the `startFrame(int)` method.

Throws:

`java.lang.IllegalStateException` - If this component is not displayable.

See Also:

`java.awt.Component.isDisplayable()`, `startFrame(int)`

startAnimation()

```
public void startAnimation()
```

Start this animation immediately, and reset the frame number to zero. Frame zero will be the first frame that can be displayed; typically it will be one or two frames after the frame visible on the screen at the time this method is called.

Applications should only call this method on a stopped animation. However, if this method is called when an animation is already started, it is re-started; the effect is equivalent to calling `stopAnimation(true)` followed by `start-Animation()`.

See Also:

`isStarted()`

startAnimationAt(Clock, Time)

```
public void startAnimationAt(javax.media.Clock c, javax.media.Time t)
throws IncompatibleTimeBaseException
```

Start this animation keyed to the clock at the given media time. If the clock's media time is already greater than the given time, this is equivalent to `startAnimation()`. Otherwise, once the clock's media time is greater than or equal to the given the given value, the animation will be started. Callers should not assume that frame zero of the animation will coincide with the desired time in all cases; for example, the clock's media time might advance in a discontinuous manner. Callers should always consult `getMediaTime(...)`.

This method can be used to initiate frame-accurate animation that is synchronized to video that is being presented on the same screen. The animation enters the started state, and drawing to graphics buffers can begin. The system will start copying these buffers to the framebuffer automatically, when the clock reaches the given time.

Subsequent calls to this method override any previous calls. If this method is called when an animation is already started, it is re-started; the effect is equivlanet to calling `stopAnimation(true)` followed by `startAnimationAt(...)`.

Parameters:

`c` - A JMF Clock that is associated with some media.

`t` - A media time of that clock when the animation should start.

Throws:

`javax.media.IncompatibleTimeBaseException` - If this Clock is incompatible with this animation. This will never be thrown if the Clock is associated with video being displayed on the same screen as this animation.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 597 of 861

`java.lang.IllegalStateException` - If this clock is not in the started state.

See Also:

`isStarted()`, `getMediaTime(Clock, int)`

startFrame(int)

```
public int startFrame(int frameNumber)
throws InterruptedException
```

Start drawing the given frame. The return value gives the index into the array obtained from `getBuffersGraphics()` for drawing of this frame, or -1 if the animation has fallen behind, and a later frame should now be drawn. After calling this method, if a value other than -1 is returned, the caller may draw to the indicated graphics buffer. When it is finished, it shall call `finishFrame()`.

If a buffer is not available for the given frame, this method will block until one is ready.

The caller can always skip frames. For example, a caller wishing to animate at half of the component's framerate could request frames 0, 2, 4, 6, 8, 10, etc. In this example, if there are four buffers and animation does not fall behind, the caller would be instructed to draw into buffer 0, 1, 2, 3, 0, 1, etc. A caller that wishes to start animating at a frame greater than 0 may do so by simply starting with a number greater than zero; when the lower-numbered frames are being presented, the component will simply do no drawing.

The content of the framebuffers is not modified by the system. Thus, a caller that is drawing into buffer number *n* could function correctly if it only drew to pixels that have changed since it last drew into buffer number *n*.

Parameters:

`frameNumber` - The frame number to draw. The first frame is frame 0.

Returns:

An index into the array of graphics objects for drawing the given frame, or -1 if animation has fallen behind, and a later frame should now be drawn.

Throws:

`java.lang.IllegalArgumentException` - if frameNumber is less than or equal to a number previously supplied to this animation, or is less than zero.

`java.lang.IllegalStateException` - if `startFrame()` has already been successfully called without a corresponding `finishFrame()`.

`java.lang.InterruptedException` - If this animation is in the stopped state, either when this method is called or due to a state transition while it is blocked waiting for a graphics buffer.

See Also:

`getBuffersGraphics()`, `isStarted()`

stopAnimation(boolean)

```
public void stopAnimation(boolean immediate)
```

Stops this animation. If it is already in the stopped state, this method has no effect. If it is in the started state, it is set to the stopped state. Once this component is in the stopped state, it will not draw into any pixels to the screen, or as a result of a call to the `paint()` method.

The caller may request that queued fames of animation be output to the screen. This will be done if the animation is in the started state, and immediate is set false.

If a successful call to `startFrame(int)` has not yet been matched with a call to `finishFrame(int)`, this object is set to the stopped state, which will cause `finishFrame(int)` ti fail. See that method for details.

After this method returns, `isStarted()` will return false. If this animation is not in the started state, calling this method will have no effect.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 598 of 861

Parameters:

`immediate` - If the component should immediately stop copying buffers to the screen, instead of letting any queued animation frames be output.

See Also:

`isStarted()`, `finishFrame(int)`

org.dvb.ui

DVBAlphaComposite

Declaration

```
public final class DVBAlphaComposite

java.lang.Object
  |
  +--org.dvb.ui.DVBAlphaComposite
```

Description

This `DVBAlphaComposite` class implements the basic alpha compositing rules for combining source and destination pixels to achieve blending and transparency effects with graphics, images and video. The rules implemented by this class are a subset of the Porter-Duff rules described in T. Porter and T. Duff, "Compositing Digital Images", SIGGRAPH 84, 253-259.

If any input does not have an alpha channel, an alpha value of 1,0, which is completely opaque, is assumed for all pixels. A constant alpha value can also be specified to be multiplied with the alpha value of the source pixels.

The following abbreviations are used in the description of the rules:

- Cs = one of the color components of the source pixel without alpha.
- cs = color component of a source pixel pre-multiplied with alpha (cs = As*Ar*Cs).
- Cd = one of the color components of the destination pixel without alpha.
- cd = color component of a destination pixel pre-multiplied with alpha
- Cn = the new constructed color without alpha.
- cn = the new constructed color pre-multiplied with alpha.
- As = alpha component of the source pixel.
- Ad = alpha component of the destination pixel.
- An = the new alpha after compositing.
- Ar = alpha, specified by getInstance(int Rule, float Ar). Unless otherwise specified Ar = 1,0f.
- Fs = fraction of the source pixel that contributes to the output.
- Fd = fraction of the input destination pixel that contributes to the output.

The color and alpha components produced by the compositing operation are calculated as follows:

```
cn = (As*Ar)*Cs*Fs + Ad*Cd*Fd
An = (As*Ar)*Fs + Ad*Fd
 Cn = cn/An
```

where Fs and Fd are specified by each rule.

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

The alpha resulting from the compositing operation is stored in the destination if the destination has an alpha channel. Otherwise, the resulting color is divided by the resulting alpha before being stored in the destination and the alpha is discarded. If the alpha value is 0,0, the color values are set to 0,0.

See Also:

`java.awt.AlphaComposite`

Fields

Clear

```
public static final org.dvb.ui.DVBAlphaComposite Clear
```

`DVBAlphaComposite` object that implements the opaque CLEAR rule with an alpha (Ar) of 1,0f.

See Also:

`CLEAR`

CLEAR

```
public static final int CLEAR
```

Porter-Duff Clear rule. Both the color and the alpha of the destination are cleared. Neither the source nor the destination is used as input.

Fs = 0 and Fd = 0, thus:

```
cn = 0
An = 0
 Cn = 0
```

Note that this operation is a fast drawing operation This operation is the same as using a source with alpha= 0 and the SRC rule.

DST_IN

```
public static final int DST_IN
```

Porter-Duff Destination In Source rule. The part of the destination lying inside of the source replaces the destination.

Fs = 0 and Fd = (As*Ar), thus:

```
cn = Ad*Cd*(As*Ar)
An = Ad*(As*Ar)
 Cn = Cd
```


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

Note that this operation is faster than e.g. SRC_OVER but slower than SRC

DST_OUT

```
public static final int DST_OUT
```

Porter-Duff Destination Held Out By Source rule. The part of the destination lying outside of the source replaces the destination.

Fs = 0 and Fd = (1-(As*Ar)), thus:

```
cn = Ad*Cd*(1-(As*Ar))
An = Ad*(1-(As*Ar))
 Cn = Cd
```

Note that this operation is faster than e.g. SRC_OVER but slower than SRC

DST_OVER

```
public static final int DST_OVER
```

Porter-Duff Destination Over Source rule. The destination is composited over the source and the result replaces the destination.

Fs = (1-Ad) and Fd = 1, thus:

```
cn = (As*Ar)*Cs*(1-Ad) + Ad*Cd
An = (As*Ar)*(1-Ad) + Ad
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 601 of 861

Note that this can be a very slow drawing operation

DstIn

```
public static final org.dvb.ui.DVBAlphaComposite DstIn
```

`DVBAlphaComposite` object that implements the opaque DST_IN rule with an alpha (Ar) of 1,0f.

See Also:

`DST_IN`

DstOut

```
public static final org.dvb.ui.DVBAlphaComposite DstOut
```

`DVBAlphaComposite` object that implements the opaque DST_OUT rule with an alpha (Ar) of 1,0f.

See Also:

`DST_OUT`

DstOver

```
public static final org.dvb.ui.DVBAlphaComposite DstOver
```

`DVBAlphaComposite` object that implements the opaque DST_OVER rule with an alpha (Ar) of 1,0f.

See Also:

`DST_OVER`

Src

```
public static final org.dvb.ui.DVBAlphaComposite Src
```

`DVBAlphaComposite` object that implements the opaque SRC rule with an alpha (Ar) of 1,0f.

See Also:

`SRC`

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 602 of 861

SRC

```
public static final int SRC
```

Porter-Duff Source rule. The source is copied to the destination. The destination is not used as input.

Fs = 1 and Fd = 0, thus:

```
cn = (As*Ar)*Cs
An = As*Ar
 Cn = Cs
```

Note that this is a fast drawing routine

SRC_IN

```
public static final int SRC_IN
```

Porter-Duff Source In Destination rule. The part of the source lying inside of the destination replaces the destination.

Fs = Ad and Fd = 0, thus:

```
cn = (As*Ar)*Cs*Ad
An = (As*Ar)*Ad
 Cn = Cs
```

Note that this operation is faster than e.g. SRC_OVER but slower than SRC

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 603 of 861

SRC_OUT

```
public static final int SRC_OUT
```

Porter-Duff Source Held Out By Destination rule. The part of the source lying outside of the destination replaces the destination.

Fs = (1-Ad) and Fd = 0, thus:

```
cn = (As*Ar)*Cs*(1-Ad)
An = (As*Ar)*(1-Ad)
 Cn = Cs
```

Note that this operation is faster than e.g. SRC_OVER but slower than SRC

SRC_OVER

```
public static final int SRC_OVER
```

Porter-Duff Source Over Destination rule. The source is composited over the destination.

Fs = 1 and Fd = (1-(As*Ar)), thus:

```
cn = (As*Ar)*Cs + Ad*Cd*(1-(As*Ar))
An = (As*Ar) + Ad*(1-(As*Ar))
```

Note that this can be a very slow drawing operation

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 604 of 861

SrcIn

```
public static final org.dvb.ui.DVBAlphaComposite SrcIn
```

DVBAlphaComposite object that implements the opaque SRC_IN rule with an alpha (Ar) of 1,0f.

See Also:

SRC_IN

SrcOut

```
public static final org.dvb.ui.DVBAlphaComposite SrcOut
```

DVBAlphaComposite object that implements the opaque SRC_OUT rule with an alpha (Ar) of 1,0f.

See Also:

SRC_OUT

SrcOver

```
public static final org.dvb.ui.DVBAlphaComposite SrcOver
```

DVBAlphaComposite object that implements the opaque SRC_OVER rule with an alpha (Ar) of 1,0f.

See Also:

SRC_OVER

Methods

equals(Object)

```
public boolean equals(java.lang.Object obj)
```

Tests if the specified java.lang.Object is equal to this DVBAlphaComposite object.

Overrides:

equals in class Object

Parameters:

obj - the Object to test for equality.

Returns:

true if obj is a DVBAlphaComposite and has the same values for rule and alpha as this object. Otherwise false shall be returned.

getAlpha()

```
public float getAlpha()
```

Returns the alpha value of this DVBAlphaComposite. If this DVBAlphaComposite does not have an alpha value, 1,0 is returned.

Returns:

the alpha value of this DVBAlphaComposite.

getInstance(int)

```
public static org.dvb.ui.DVBAlphaComposite getInstance(int rule)
```

Creates an DVBAlphaComposite object with the specified rule. The value for alpha shall be 1,0f.

Parameters:

rule - the compositing rule.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 605 of 861

Returns:

an DVBAlphaComposite object with the specified rule.

getInstance(int, float)

```
public static org.dvb.ui.DVBAlphaComposite getInstance(int rule, float alpha)
```

Creates an DVBAlphaComposite object with the specified rule and the constant alpha (Ar) to multiply with the alpha of the source (As). The source is multiplied with the specified alpha before being composited with the destination.

Parameters:

rule - the compositing rule.

alpha - the constant alpha (Ar) to be multiplied with the alpha of the source (As). alpha must be a floating point number in the inclusive range [0,0, 1,0].

Returns:

an DVBAlphaComposite object with the specified rule and the constant alpha to multiply with the alpha of the source.

getRule()

```
public int getRule()
```

Returns the compositing rule of this DVBAlphaComposite.

Returns:

the compositing rule of this DVBAlphaComposite.

org.dvb.ui

DVBBufferedImage

Declaration

```
public class DVBBufferedImage extends java.awt.Image
```

```
java.lang.Object
  |
  +--java.awt.Image
        |
        +--org.dvb.ui.DVBBufferedImage
```

Description

The DVBBufferedImage subclass describes an java.awt.Image with an accessible buffer of image data. The DVBBufferedImage is an adapter class for java.awt.image.BufferedImage. It supports two different platform dependent sample models TYPE_BASE and TYPE_ADVANCED. Buffered images with the TYPE_BASE have the same sample model as the on screen graphics buffer, thus TYPE_BASE could be CLUT based. TYPE_ADVANCED has a direct color model but it is not specified how many bits are used to store the different color components. By default, a new DVBBufferedImage is transparent. All alpha values are set to 0; Instances of DVBBufferedImage shall be considered to be off-screen images for the purpose of the inherited method Image.getGraphics.

Since:

MHP 1.0.

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

Fields

TYPE_ADVANCED

```
public static final int TYPE_ADVANCED
```

Represents an image stored in a best possible SampleModel (platform dependent) The image has a DirectColorModel with alpha. The color data in this image is considered not to be pre-multiplied with alpha. The data returned by getRGB() will be in the TYPE_INT_ARGB color model that is alpha component in bits 24 to 31, the red component in bits 16 to 23, the green component in bits 8 to 15, and the blue component in bits 0 to 7. The data for setRGB() shall be in the TYPE_INT_ARGB color model as well.

Since:

MHP 1.0.

TYPE_BASE

```
public static final int TYPE_BASE
```

Represents an image stored in a platform dependent Sample Model. This color model is not visible to applications. The data returned by getRGB() will be in the TYPE_INT_ARGB color model that is alpha component in bits 24 to 31, the red component in bits 16 to 23, the green component in bits 8 to 15, and the blue component in bits 0 to 7. The data for setRGB() shall be in the TYPE_INT_ARGB color model as well.

Since:

MHP 1.0.

Constructors

DVBBufferedImage(int, int)

```
public DVBBufferedImage(int width, int height)
```

Constructs a DVBBufferedImage with the specified width and height. The Sample Model used the image is the native Sample Model (TYPE_BASE) of the implementation. Note that a request can lead to an java.lang.OutOfMemoryError. Applications should be aware of this.

Parameters:

`width` - the width of the created image.

`height` - the height of the created image.

Since:

MHP 1.0.

DVBBufferedImage(int, int, int)

```
public DVBBufferedImage(int width, int height, int type)
```

Constructs a new DVBBufferedImage with the specified width and height in the Sample Model specified by type. Note that a request can lead to an java.lang.OutOfMemoryError. Applications should be aware of this.

Parameters:

`width` - the width of the DVBBufferedImage.

`height` - the height of the DVBBufferedImage.

`type` - the ColorSpace of the DVBBufferedImage.

Since:

MHP 1.0.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 607 of 861

Methods

createGraphics()

```
public org.dvb.ui.DVBGraphics createGraphics()
```

Creates a `DVBGraphics`, which can be used to draw into this `DVBBufferedImage`. Calls to this method after calls to the `dispose` method on the same instance shall return null.

Returns:

a `DVBGraphics`, used for drawing into this image.

Since:

MHP 1.0.

dispose()

```
public void dispose()
```

Disposes of this buffered image. This method releases the resources (e.g. pixel memory) underlying this buffered image. After calling this method:

- the image concerned may not be used again;
- the image shall be considered to have a width and height of -1, -1 as specified for instances of java.awt.Image where the width and height are not yet known;
- the `getGraphics` method may return null.

Since:

MHP 1.0.1.

flush()

```
public void flush()
```

Flushes all resources being used to cache optimization information. The underlying pixel data is unaffected. Calls to this method after calls to the `dispose` method on the same instance shall fail silently.

Overrides:

`flush` in class `Image`

getGraphics()

```
public java.awt.Graphics getGraphics()
```

This method returns a `java.awt.Graphics` , it is here for backwards compatibility. `createGraphics` is more convenient, since it is declared to return a `DVBGraphics`. Calls to this method after calls to the `dispose` method on the same instance shall return null.

Overrides:

getGraphics in class Image

Returns:

a `Graphics`, which can be used to draw into this image.

getHeight()

```
public int getHeight()
```

Returns the height of the `DVBBufferedImage`.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 608 of 861

Returns:

the height of this DVBBufferedImage.

Since:

MHP 1.0.

getHeight(ImageObserver)

```
public int getHeight(java.awt.image.ImageObserver observer)
```

Returns the height of the image.

Overrides:

getHeight in class Image

Parameters:

observer - the ImageObserver that receives information about the image.

Returns:

the height of the image or -1 if the height is not yet known.

See Also:

java.awt.Image.getWidth(ImageObserver), java.awt.image.ImageObserver

getImage()

```
public java.awt.Image getImage()
```

Returns a java.awt.Image representing this buffered image. In implementations which implement java.awt.image.BufferedImage this returns a java.awt.image.BufferedImage cast to a java.awt.Image. Otherwise it is implementation dependent whether it returns this image or whether it returns an instance of an underlying platform specific sub-class of java.awt.Image. Calls to this method after calls to the dispose method on the same instance shall return null.

Returns:

a java.awt.image representing this buffered image.

Since:

MHP 1.0.

getProperty(String, ImageObserver)

```
public java.lang.Object getProperty(java.lang.String name, java.awt.image.ImageObserver observer)
```

Returns a property of the image by name. Individual property names are defined by the various image formats. If a property is not defined for a particular image, this method returns the UndefinedProperty field. If the properties for this image are not yet known, then this method returns null and the ImageObserver object is notified later. The property name "comment" should be used to store an optional comment that can be presented to the user as a description of the image, its source, or its author. Calls to this method after calls to the dispose method on the same instance shall return null.

Overrides:

getProperty in class Image

Parameters:

name - the property name.

observer - the ImageObserver that receives notification regarding image information.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 609 of 861

Returns:

an `java.lang.Object` that is the property referred to by the specified `name` or `null` if the properties of this image are not yet known.

See Also:

`java.awt.image.ImageObserver`, `java.awt.Image.UndefinedProperty`

getRGB(int, int)

```
public int getRGB(int x, int y)
```

Returns the specified integer pixel in the default RGB color model (TYPE_INT_ARGB) and default sRGB colorspace. Color conversion takes place if the used Sample Model is not 8-bit for each color component There are only 8-bits of precision for each color component in the returned data when using this method. Note that when a lower precision is used in this buffered image getRGB may return different values than those used in setRGB()

Parameters:

`x` - the x-coordinate of the pixel.

`y` - the y-coordinate of the pixel.

Returns:

an integer pixel in the default RGB color model (TYPE_INT_ARGB) and default sRGB colorspace.

Throws:

`java.lang.ArrayIndexOutOfBoundsException` - if x or y is out of bounds or if the dispose method has been called on this instance.

Since:

MHP 1.0.

getRGB(int, int, int, int, int[], int, int)

```
public int[] getRGB(int startX, int startY, int w, int h, int[] rgbArray, int offset, int scansize)
```

Returns an array of integer pixels in the default RGB color model (TYPE_INT_ARGB) and default sRGB color space, from a rectangular region of the image data. There are only 8-bits of precision for each color component in the returned data when using this method. With a specified coordinate (x, y) in the image, the ARGB pixel can be accessed in this way:

```
pixel   = rgbArray[offset + (y-startY)*scansize + (x-startX)];
```

Parameters:

`startX` - the x-coordinate of the upper-left corner of the specified rectangular region.

`startY` - the y-coordinate of the upper-left corner of the specified rectangular region.

`w` - the width of the specified rectangular region.

`h` - the height of the specified rectangular region.

`rgbArray` - if not `null`, the rgb pixels are written here.

`offset` - offset into the `rgbArray`.

`scansize` - scanline stride for the `rgbArray`.

Returns:

array of ARGB pixels.

Throws:

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 610 of 861

java.lang.ArrayIndexOutOfBoundsException - if the specified portion of the image data is out of bounds or if the dispose method has been called on this instance.

Since:

MHP 1.0.

getScaledInstance(int, int, int)

```
public java.awt.Image getScaledInstance(int width, int height, int hints)
```

Creates a scaled version of this image. A new Image object is returned which will render the image at the specified width and height by default. The new Image object may be loaded asynchronously even if the original source image has already been loaded completely. If either the width or height is a negative number then a value is substituted to maintain the aspect ratio of the original image dimensions. If the dispose method has been called on this instance then null shall be returned.

Overrides:

getScaledInstance in class Image

Parameters:

width - the width to which to scale the image.

height - the height to which to scale the image.

hints - flags to indicate the type of algorithm to use for image resampling.

Returns:

a scaled version of the image.

getSource()

```
public java.awt.image.ImageProducer getSource()
```

Returns the object that produces the pixels for the image.

Overrides:

getSource in class Image

Returns:

the java.awt.image.ImageProducer that is used to produce the pixels for this image.

If the dispose method has been called on this instance then null shall be returned. The source returned by this method is platform generated to provide access to the current contents of the DVBBufferedImage buffer.

See Also:

java.awt.image.ImageProducer

getSubimage(int, int, int, int)

```
public org.dvb.ui.DVBBufferedImage getSubimage(int x, int y, int w, int h)
throws DVBRasterFormatException
```

Returns a subimage defined by a specified rectangular region. The returned DVBBufferedImage shares the same data array as the original image. If the dispose method has been called on this instance then null shall be returned.

Parameters:

x - the x-coordinate of the upper-left corner of the specified rectangular region.

y - the y-coordinate of the upper-left corner of the specified rectangular region.

w - the width of the specified rectangular region.

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

h - the height of the specified rectangular region.

Returns:

a DVBBufferedImage that is the subimage of this DVBBufferedImage.

Throws:

DVBRasterFormatException - if the specified area is not contained within this DVBBufferedImage.

Since:

MHP 1.0.

getWidth()

```
public int getWidth()
```

Returns the width of the DVBBufferedImage.

Returns:

the width of this DVBBufferedImage.

Since:

MHP 1.0.

getWidth(ImageObserver)

```
public int getWidth(java.awt.image.ImageObserver observer)
```

Returns the width of the image. If the width is not known yet then the java.awt.image.ImageObserver is notified later and -1 is returned.

Overrides:

getWidth in class Image

Parameters:

observer - the ImageObserver that receives information about the image.

Returns:

the width of the image or -1 if the width is not yet known.

See Also:

java.awt.Image.getHeight(ImageObserver), java.awt.image.ImageObserver

setRGB(int, int, int)

```
public void setRGB(int x, int y, int rgb)
```

Sets a pixel in this DVBBufferedImage to the specified ARGB value. The pixel is assumed to be in the default RGB color model, TYPE_INT_ARGB, and default sRGB color space. Calls to this method after calls to the dispose method on the same instance shall throw an ArrayIndexOutOfBoundsException.

Parameters:

x - the x-coordinate of the pixel to set.

y - the y-coordinate of the pixel to set.

rgb - the ARGB value.

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

Since:

MHP 1.0.

setRGB(int, int, int, int, int[], int, int)

```
public void setRGB(int startX, int startY, int w, int h, int[] rgbArray, int offset, int scansize)
```

Sets an array of integer pixels in the default RGB color model (TYPE_INT_ARGB) and default sRGB color space, into a rectangular portion of the image data. There are only 8-bits of precision for each color component in the returned data when using this method. With a specified coordinate (x, y) in the this image, the ARGB pixel can be accessed in this way:

```
pixel   = rgbArray[offset + (y-startY)*scansize + (x-startX)];
```

WARNING: No dithering takes place.

Calls to this method after calls to the `dispose` method on the same instance shall throw an `ArrayIndexOutOfBounds-Exception`.

Parameters:

`startX` - the x-coordinate of the upper-left corner of the specified rectangular region.

`startY` - the y-coordinate of the upper-left corner of the specified rectangular region.

`w` - the width of the specified rectangular region.

`h` - the height of the specified rectangular region.

`rgbArray` - the ARGB pixels.

`offset` - offset into the `rgbArray`.

`scansize` - scanline stride for the `rgbArray`.

Since:

MHP 1.0.

toString()

```
public java.lang.String toString()
```

Returns a `String` representation of this `DVBBufferedImage` object and its values.

Overrides:

`toString` in class `Object`

Returns:

a `String` representing this `DVBBufferedImage`.

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

org.dvb.ui

DVBColor

Declaration

```
public class DVBColor extends javax.tv.graphics.AlphaColor

java.lang.Object
  |
  +--java.awt.Color
        |
        +--javax.tv.graphics.AlphaColor
              |
              +--org.dvb.ui.DVBColor
```

All Implemented Interfaces:

```
java.io.Serializable, java.awt.Transparency
```

Description

A Color class which adds the notion of alpha. Because DVBColor extends Color the signatures in the existing classes do not change. Classes like Component should work with DVBColor internally. Instances of this class are a container for the values which are passed in to the constructor. Any approximations made by the platform are made when the colors are used. Note: org.dvb.ui.DVBColor adds support for alpha (compared to JDK1.1.8) and is intended to be compatible with the JDK1.2 java.awt.Color class - since org.dvb.ui.DVBColor extends javax.tv.graphics.AlphaColor which in turn extends java.awt.Color. In implementations where java.awt.Color supports alpha, such as JDK1.2, etc., the alpha-related methods in org.dvb.ui.DVBColor could just call super.

Since:

MHP 1.0.

Constructors

DVBColor(Color)

```
public DVBColor(java.awt.Color c)
```

Constructs a new DVBColor using the specified color. If c supports alpha, e.g. if it is an instance of javax.tv.graphics.AlphaColor or JDK 1.2's java.awt.Color, then the alpha value of c shall be used. If this color has no alpha value, alpha will be set to 255 (opaque).

Parameters:

`c` - the java.awt.Color used to create a new DVBColor.

DVBColor(float, float, float, float)

```
public DVBColor(float r, float g, float b, float a)
```

Creates an sRGB color with the specified red, green, blue, and alpha values in the range (0,0 to 1,0). The actual color used in rendering will depend on finding the best match given the color space available for a given output device.

Parameters:

`r` - the red component.

`g` - the green component.

`b` - the blue component.

`a` - the alpha component.

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

See Also:

java.awt.Color.getRed(), java.awt.Color.getGreen(), java.awt.Color.getBlue(), getAlpha(), getRGB()

DVBColor(int, boolean)

```
public DVBColor(int rgba, boolean hasalpha)
```

Creates an sRGB color with the specified combined RGBA value consisting of the alpha component in bits 24 to 31, the red component in bits 16 to 23, the green component in bits 8 to 15, and the blue component in bits 0 to 7. If the hasalpha argument is False, alpha is defaulted to 255.

Parameters:

`rgba` - the combined RGBA components.

`hasalpha` - true if the alpha bits are valid, false otherwise.

See Also:

java.awt.Color.getRed(), java.awt.Color.getGreen(), java.awt.Color.getBlue(), getAlpha(), getRGB()

DVBColor(int, int, int, int)

```
public DVBColor(int r, int g, int b, int a)
```

Creates an sRGB color with the specified red, green, blue, and alpha values in the range (0 to 255).

Parameters:

`r` - the red component.

`g` - the green component.

`b` - the blue component.

`a` - the alpha component.

See Also:

java.awt.Color.getRed(), java.awt.Color.getGreen(), java.awt.Color.getBlue(), getAlpha(), getRGB()

Methods

brighter()

```
public java.awt.Color brighter()
```

Creates a brighter version of this color. This method applies an arbitrary scale factor to each of the three RGB components of the color to create a brighter version of the same color. Although brighter and darker are inverse operations, the results of a series of invocations of these two methods may be inconsistent because of rounding errors. The alpha value shall be preserved.

Overrides:

`brighter` in class `AlphaColor`

Returns:

a new DVBColor object (cast to a java.awt.Color object) representing a brighter version of this color. Applications can recast it to a org.dvb.ui.DVBColor object.

See Also:

java.awt.Color.brighter()

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

darker()

```
public java.awt.Color darker()
```

Creates a darker version of this color. This method applies an arbitrary scale factor to each of the three RGB components of the color to create a darker version of the same color. Although brighter and darker are inverse operations, the results of a series of invocations of these two methods may be inconsistent because of rounding errors. The alpha value shall be preserved.

Overrides:

`darker` in class `AlphaColor`

Returns:

a new DVBColor object (cast to a java.awt.Color object), representing a darker version of this color. Applications can recast it to a org.dvb.ui.DVBColor object.

See Also:

`java.awt.Color.darker()`

equals(Object)

```
public boolean equals(java.lang.Object obj)
```

Determines whether another object is equal to this color. The result is true if and only if the argument is not null and is a DVBColor object that has the same red, green, blue and alpha values as this object.

Overrides:

`equals` in class `AlphaColor`

Parameters:

`obj` - - the object to compare with.

Returns:

true if the objects are the same; false otherwise.

Since:

MHP 1.0.

getAlpha()

```
public int getAlpha()
```

Returns the alpha component. In the range 0 to 255.

Overrides:

`getAlpha` in class `AlphaColor`

Returns:

the alpha component.

See Also:

`getRGB()`

getRGB()

```
public int getRGB()
```

Returns the RGB value representing the color in the default sRGB ColorModel. (Bits 24 to 31 are alpha, 16 to 23 are red, 8 to 15 are green, 0 to 7 are blue).

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 616 of 861

Overrides:

getRGB in class AlphaColor

Returns:

the RGB value representing the color in the default sRGB ColorModel.

Since:

MHP 1.0.

See Also:

java.awt.Color.getRed(), java.awt.Color.getGreen(), java.awt.Color.getBlue(), getAlpha()

toString()

```
public java.lang.String toString()
```

Creates a string that represents this color and indicates the values of its ARGB components.

Overrides:

toString in class AlphaColor

Returns:

a representation of this color as a String object.

Since:

MHP 1.0

org.dvb.ui

DVBGraphics

Declaration

```
public abstract class DVBGraphics extends java.awt.Graphics2D
```

```
java.lang.Object
  |
  +--java.awt.Graphics
        |
        +--java.awt.Graphics2D
              |
              +--org.dvb.ui.DVBGraphics
```

Description

The DVBGraphics class is a adapter class to support alpha compositing in a GEM device. At all times, the DVBAlpha-Composite returned by getDVBComposite and the Composite returned by the getComposite method inherited from Graphics2D shall be consistent. This means that the results of calling getRule and getAlpha on those shall be identical. In GEM devices all Graphics Objects are DVBGraphics objects. Thus one can get a DVBGraphics by casting a given Graphics object. The normal compositing rule used is DVBAlphaComposite.SRC_OVER. Note that the default rule of SRC_OVER may not give the highest performance. Under many circumstances, applications will find that the SRC rule will give higher performance. The intersection between setDVBComposite in this class and the setPaintMode and setXORMode methods inherited from java.awt.Graphics shall be as follows:

- Calling setPaintMode on an instance of this class shall be equivalent to calling set-DVBComposite(DVBAlphaComposite.SrcOver).
- The present document does not tighten, refine or detail the definition of the setXORMode beyond what is specified for the parent class.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 617 of 861

NOTE: Implementations of XOR mode may change colours with alpha to without and vice versa (reversibly).

Since:

MHP1.0.

See Also:

`java.awt.Graphics`

Constructors

DVBGraphics()

```
protected DVBGraphics()
```

Constructs a new `DVBGraphics` object. This constructor is the default constructor for a graphics context.

Since `DVBGraphics` is an abstract class, applications cannot call this constructor directly. DVBGraphics contexts are obtained from other DVBGraphics contexts or are created by casting java.awt.Graphics to DVBGraphics.

Since:

MHP 1.0.

See Also:

`java.awt.Graphics.create(), java.awt.Component.getGraphics()`

Methods

getAvailableCompositeRules()

```
public abstract int[] getAvailableCompositeRules()
```

Returns all available Porter-Duff Rules for this specific Graphics context. E.g. a devices could support the SRC_OVER rule when using a destination which does not have Alpha or where the alpha is null, while this rule is not available when drawing on a graphic context where the destination has alpha. Which rules are supported for the different graphics objects is defined in the Minimum Platform Capabilities of the GEM spec.

Returns:

all available Porter-Duff Rules for this specific Graphics context.

Since:

MHP 1.0.

getBestColorMatch(Color)

```
public org.dvb.ui.DVBColor getBestColorMatch(java.awt.Color c)
```

Returns the best match for the specified Color as a DVBColor, in a device-dependent manner, as constrained by the GEM graphics reference model.

Parameters:

`c` - the specified Color.

Returns:

the best DVBColor match for the specified Color.

Since:

MHP 1.0.

getColor()

```
public abstract java.awt.Color getColor()
```


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

Gets this graphics context's current color. This will return a DVBColor cast to java.awt.Color.

Overrides:

`getColor` in class `Graphics`

Returns:

this graphics context's current color.

Since:

MHP 1.0.

See Also:

`DVBColor`, `java.awt.Color`, `setColor(Color)`

getDVBComposite()

```
public abstract org.dvb.ui.DVBAlphaComposite getDVBComposite()
```

Returns the current `DVBAlphaComposite` in the `DVBGraphics` context. This method could delegate to a java.awt.Graphics2D object where available

Returns:

the current `DVBGraphics` `DVBAlphaComposite`, which defines a compositing style.

Since:

MHP 1.0.

See Also:

`setDVBComposite(DVBAlphaComposite)`

getType()

```
public int getType()
```

Returns the Sample Model (DVBBufferedImage.TYPE_BASE, DVBBufferedImage.TYPE_ADVANCED) which is used in the on/off screen buffer this graphics object draws into.

Returns:

the type of the Sample Model.

Since:

MHP 1.0.

See Also:

`DVBBufferedImage`

setColor(Color)

```
public abstract void setColor(java.awt.Color c)
```

Sets this graphics context's current color to the specified color. All subsequent graphics operations using this graphics context use this specified color. Note that color c can be a DVBColor.

Overrides:

`setColor` in class `Graphics`

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 619 of 861

Parameters:

c - the new rendering color.

Since:

MHP 1.0.

See Also:

java.awt.Color, DVBColor, getColor()

setDVBComposite(DVBAlphaComposite)

```
public abstract void setDVBComposite(org.dvb.ui.DVBAlphaComposite comp)
throws UnsupportedDrawingOperationException
```

Sets the DVBAlphaComposite for the DVBGraphics context. The DVBAlphaComposite is used in all drawing methods such as drawImage, drawString, draw, and fill. It specifies how new pixels are to be combined with the existing pixels on the graphics device during the rendering process.

This method could delegate to a Graphics2D object or to an native implementation

Parameters:

comp - the DVBAlphaComposite object to be used for rendering.

Throws:

UnsupportedDrawingOperationException - when the requested Porter-Duff rule is not supported by this graphics context.

Since:

MHP 1.0.

See Also:

java.awt.Graphics.setXORMode(Color), java.awt.Graphics.setPaintMode(), DVBAlphaComposite

toString()

```
public java.lang.String toString()
```

Returns a String object representing this DVBGraphics object's value.

Overrides:

toString in class Graphics

Returns:

a string representation of this graphics context.

Since:

MHP 1.0.

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

org.dvb.ui DVBRasterFormatException

Declaration

```
public class DVBRasterFormatException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.ui.DVBRasterFormatException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This exception is thrown for some invalid operations on instances of `DVBBufferedImage`. The precise conditions are defined in the places where this exception is thrown.

Since:

MHP 1.0.1.

See Also:

```
DVBBufferedImage
```

Constructors

DVBRasterFormatException(String)

```
public DVBRasterFormatException(java.lang.String s)
```

Constructs an instance of `DVBRasterFormatException` with the specified detail message.

Parameters:

`s` - the detail message.

Since:

MHP1.0.

org.dvb.ui DVBTextLayoutManager

Declaration

```
public class DVBTextLayoutManager implements org.havi.ui.HTextLayoutManager

java.lang.Object
  |
  +--org.dvb.ui.DVBTextLayoutManager
```

All Implemented Interfaces:

```
org.havi.ui.HTextLayoutManager
```


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

Description

The DVBTextLayoutManager provides a text rendering layout mechanism for the org.havi.ui.HStaticText org.havi.ui.HText and org.havi.ui.HTextButton classes.

The semantics of the rendering behaviour and the settings are specified in the "Text presentation" annex of the present document. The DVBTextLayoutManager renders the text according to the semantics described in that annex.

Fields

HORIZONTAL_CENTER

```
public static final int HORIZONTAL_CENTER
```

The text should be centered horizontally.

HORIZONTAL_END_ALIGN

```
public static final int HORIZONTAL_END_ALIGN
```

The text should be horizontally to the horizontal end side (e.g. when start corner is upper left and line orientation horizontal, meaning text that is read left to right from top to bottom, this implies alignment to right).

HORIZONTAL_START_ALIGN

```
public static final int HORIZONTAL_START_ALIGN
```

The text should be aligned horizontally to the horizontal start side (e.g. when start corner is upper left and line orientation horizontal, meaning text that is read left to right from top to bottom, this implies alignment to left).

LINE_ORIENTATION_HORIZONTAL

```
public static final int LINE_ORIENTATION_HORIZONTAL
```

Horizontal line orientation.

LINE_ORIENTATION_VERTICAL

```
public static final int LINE_ORIENTATION_VERTICAL
```

Vertical line orientation.

START_CORNER_LOWER_LEFT

```
public static final int START_CORNER_LOWER_LEFT
```

Lower left text start corner.

START_CORNER_LOWER_RIGHT

```
public static final int START_CORNER_LOWER_RIGHT
```

Lower right text start corner.

START_CORNER_UPPER_LEFT

```
public static final int START_CORNER_UPPER_LEFT
```

Upper left text start corner.

START_CORNER_UPPER_RIGHT

```
public static final int START_CORNER_UPPER_RIGHT
```

Upper right text start corner.

VERTICAL_CENTER

```
public static final int VERTICAL_CENTER
```

The text should be centered vertically.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 622 of 861

VERTICAL_END_ALIGN

```
public static final int VERTICAL_END_ALIGN
```

The text should be aligned vertically to the vertical end side (e.g. when start corner is upper left and line orientation horizontal, meaning text that is read left to right from top to bottom, this implies alignment to bottom).

This is defined by the clause "Vertical limits" in the "Text presentation" annex of the present document.

VERTICAL_START_ALIGN

```
public static final int VERTICAL_START_ALIGN
```

The text should be aligned vertically to the vertical start side (e.g. when start corner is upper left and line orientation horizontal, meaning text that is read left to right from top to bottom, this implies alignment to top).

This is defined by the clause "Vertical limits" in the "Text presentation" annex of the present document.

Constructors

DVBTextLayoutManager()

```
public DVBTextLayoutManager()
```

Constructs a DVBTextLayoutManager object with default parameters (HORIZONTAL_START_ALIGN, VERTICAL_START_ALIGN, LINE_ORIENTATION_HORIZONTAL, START_CORNER_UPPER_LEFT, wrap = true, linespace = (point size of the default font for HVisible) + 7, letterspace = 0, horizontalTabSpace = 56)

DVBTextLayoutManager(int, int, int, int, boolean, int, int, int)

```
public DVBTextLayoutManager(int horizontalAlign, int verticalAlign, int lineOrientation,
int startCorner, boolean wrap, int linespace, int letterspace, int horizontalTabSpace)
```

Constructs a DVBTextLayoutManager object.

Parameters:

`horizontalAlign` - Horizontal alignment setting.

`verticalAlign` - Vertical alignment setting.

`lineOrientation` - Line orientation setting.

`startCorner` - Starting corner setting.

`wrap` - Text wrapping setting.

`linespace` - Line spacing setting expressed in points.

`letterspace` - Letterspacing adjustment relative to the default letterspacing. Expressed in units of 1/256th point as the required increase in the spacing between consecutive characters. May be either positive or negative.

`horizontalTabSpace` - Horizontal tabulation setting in points.

Methods

addTextOverflowListener(TextOverflowListener)

```
public void addTextOverflowListener(org.dvb.ui.TextOverflowListener l)
```

Register a TextOverflowListener that will be notified if the text string does not fit in the component when rendering.

Parameters:

`l` - a listener object.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 623 of 861

getHorizontalAlign()

```
public int getHorizontalAlign()
```

Get the horizontal alignment.

Returns:

Horizontal alignment setting.

getHorizontalTabSpacing()

```
public int getHorizontalTabSpacing()
```

Get the horizontal tabulation spacing.

Returns:

the horizontal tabulation spacing.

getInsets()

```
public java.awt.Insets getInsets()
```

Returns the insets set by the setInsets method. These Insets are added to the ones passed to the `render` method for rendering the text. When not previously set, zero Insets are returned.

Returns:

Insets set by the setInsets method.

getLetterSpace()

```
public int getLetterSpace()
```

Get the letter space setting. This is a 16 bit signed integer specifying in units of 1/256th point the required increase in the spacing between consecutive characters. It corresponds to the "track" parameter in the GEM text rendering rules.

Returns:

letter space setting.

getLineOrientation()

```
public int getLineOrientation()
```

Get the line orientation.

Returns:

Line orientation setting.

getLineSpace()

```
public int getLineSpace()
```

Get the line space setting.

Returns:

line space setting or -1, if the default line spacing is determined from the size of the default font used.

getStartCorner()

```
public int getStartCorner()
```

Get the starting corner.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 624 of 861

Returns:

Starting corner setting.

getTextWrapping()

```
public boolean getTextWrapping()
```

Get the text wrapping setting.

Returns:

text wrapping setting.

getVerticalAlign()

```
public int getVerticalAlign()
```

Get the vertical alignment.

Returns:

Vertical alignment setting.

removeTextOverflowListener(TextOverflowListener)

```
public void removeTextOverflowListener(org.dvb.ui.TextOverflowListener l)
```

Removes a TextOverflowListener that has been registered previously.

Parameters:

`l` - a listener object.

render(String, Graphics, HVisible, Insets)

```
public void render(java.lang.String markedUpString, java.awt.Graphics g, org.havi.ui.HVisible v,
java.awt.Insets insets)
```

Render the string. The `HTextLayoutManager` should use the passed `HVisible` object to determine any additional information required to render the string, e.g. `Font`, `Color`, etc.

The text should be laid out in the layout area, which is defined by the bounds of the specified `HVisible`, after subtracting the insets. If the insets are `null` the full bounding rectangle is used as the area to render text into.

The `HTextLayoutManager` shall not modify the clipping rectangle of the `Graphics` object.

Specified By:

`render` in interface `HTextLayoutManager`

Parameters:

`markedUpString` - the string to render.

`g` - the graphics context, including a clipping rectangle which encapsulates the area within which rendering is permitted. If a valid insets value is passed to this method then text must only be rendered into the bounds of the widget after the insets are subtracted. If the insets value is `null` then text is rendered into the entire bounding area of the `HVisible`. It is implementation specific whether or not the renderer takes into account the intersection of the clipping rectangle in each case for optimization purposes.

`v` - the `HVisible` into which to render.

`insets` - the insets to determine the area in which to layout the text, or `null`.

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

setHorizontalAlign(int)

```
public void setHorizontalAlign(int horizontalAlign)
```

Set the horizontal alignment. The setting shall be one of `HORIZONTAL_CENTER`, `HORIZONTAL_END_ALIGN` or `HORIZONTAL_START_ALIGN`. The failure mode if other values are used is implementation dependent.

Parameters:

`horizontalAlign` - Horizontal alignment setting.

setHorizontalTabSpacing(int)

```
public void setHorizontalTabSpacing(int horizontalTabSpace)
```

Set the horizontal tabulation spacing.

Parameters:

`horizontalTabSpace` - tab spacing in points.

setInsets(Insets)

```
public void setInsets(java.awt.Insets insets)
```

Sets the insets which shall be used by this DVBTextLayoutManager to provide a "virtual margin". These shall be added to the insets passed to the `Render` method (which are to be considered as "bounds"). If this method is not called, the default insets are 0 at each edge.

Parameters:

`insets` - Insets that should be used.

setLetterSpace(int)

```
public void setLetterSpace(int letterSpace)
```

Set the letter space setting. This is a 16 bit signed integer specifying in units of 1/256th point the required increase in the spacing between consecutive characters. It corresponds to the "track" parameter in the GEM text rendering rules.

Parameters:

`letterSpace` - letter space setting.

setLineOrientation(int)

```
public void setLineOrientation(int lineOrientation)
```

Set the line orientation. The setting shall be one of `LINE_ORIENTATION_VERTICAL`, `LINE_ORIENTATION_HORIZONTAL`. The failure mode if other values are used is implementation dependent.

Parameters:

`lineOrientation` - Line orientation setting.

setLineSpace(int)

```
public void setLineSpace(int lineSpace)
```

Set the line space setting. Using -1 as the line space setting shall cause the line spacing to be set to the default value as defined for the no-argument constructor for this class.

Parameters:

`lineSpace` - line space setting.

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

setStartCorner(int)

```
public void setStartCorner(int startCorner)
```

Set the starting corner. The setting shall be one of START_CORNER_UPPER_LEFT, START_CORNER_UPPER_RIGHT, START_CORNER_LOWER_LEFT or START_CORNER_LOWER_RIGHT. The failure mode if other values are used is implementation dependent.

Parameters:

startCorner - Starting corner setting.

setTextWrapping(boolean)

```
public void setTextWrapping(boolean wrap)
```

Set the text wrapping setting.

Parameters:

wrap - Text wrapping setting.

setVerticalAlign(int)

```
public void setVerticalAlign(int verticalAlign)
```

Set the vertical alignment. The setting shall be one of VERTICAL_CENTER, VERTICAL_END_ALIGN or VERTICAL_START_ALIGN. The failure mode if other values are used is implementation dependent.

Parameters:

verticalAlign - Vertical alignment setting.

org.dvb.ui

FontFactory

Declaration

```
public class FontFactory

java.lang.Object
  |
  +--org.dvb.ui.FontFactory
```

Description

Provides a mechanism for applications to instantiate fonts that are not built into the system. The two constructors of this class allow fonts to be downloaded either through the font index file of the application or directly from a font file in the format(s) specified in the main body of the present document.

Constructors

FontFactory()

```
public FontFactory()
throws FontFormatException, IOException
```

Constructs a FontFactory for the font index file bound to this application in the application signalling. The call to the constructor is synchronous and shall block until the font index file has been retrieved or an an exception is thrown.

Throws:

FontFormatException - if there is an error in the font index file bound with the application.

java.io.IOException - if there is no font index file bound with the application, or if there is an error attempting to access the data in that file.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 627 of 861

FontFactory(URL)

```
public FontFactory(java.net.URL u)
throws IOException, FontFormatException
```

Constructs a FontFactory for the font file found at the given location. The call to the constructor is synchronous and shall block until the font file has been retrieved or an exception is thrown.

Parameters:

`u` - The location of the font file.

Throws:

`java.io.IOException` - if there is an error attempting to access the data referenced by the URL.

`java.lang.IllegalArgumentException` - if the URL is not both valid and supported.

`java.lang.SecurityException` - if access to the specified URL is denied by security policy.

`FontFormatException` - if the file at that URL is not a valid font file as specified in the main body of the present document.

Methods

createFont(String, int, int)

```
public java.awt.Font createFont(java.lang.String name, int style, int size)
throws FontNotAvailableException, FontFormatException, IOException
```

Creates a font object from the font source associated with this FontFactory. This font will remain valid even if the FontFactory is no longer reachable from application code. The name returned by Font.getName() might not be the same as the name supplied, for example, it might have a string prepended to it that identifies the source FontFactory in a platform-dependant manner. For FontFactory instances bound to the font index file of an application, the call to the method is synchronous and shall block until either an exception is thrown or any required network access has completed.

The value of the style argument must be as defined in java.awt.Font. Valid values are the following:

- java.awt.Font.PLAIN.
- java.awt.Font.BOLD.
- java.awt.Font.ITALIC.
- java.awt.Font.BOLD + java.awt.Font.ITALIC.

Parameters:

`name` - the font name.

`style` - the constant style used, such as java.awt.Font.PLAIN.

`size` - the point size of the font.

Throws:

`FontNotAvailableException` - if a font with given parameters cannot be located or created.

`java.io.IOException` - if there is an error retrieving a font from the network. Thrown only for font factory instances bound to the font index file of an application.

`java.lang.IllegalArgumentException` - if the style parameter is not in the set of valid values, or if the size parameter is zero or negative.

`FontFormatException` - if the font file is not a valid font file as specified in the main body of the present document. Thrown only for font factory instances bound to the font index file of an application.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 628 of 861

org.dvb.ui
FontFormatException

Declaration

```
public class FontFormatException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.ui.FontFormatException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown when attempt is made to read a file describing a font when the contents of that file are not valid.

Constructors

FontFormatException()

```
public FontFormatException()
```

Constructs a `FontFormatException` with `null` as its error detail message.

FontFormatException(String)

```
public FontFormatException(java.lang.String s)
```

Constructs a `FontFormatException` with the specified detail message. The error message string `s` can later be retrieved by the `java.lang.Throwable.getMessage()` method of class `java.lang.Throwable`.

Parameters:

`s` - the detail message.

org.dvb.ui
FontNotAvailableException

Declaration

```
public class FontNotAvailableException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.ui.FontNotAvailableException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown when attempt is made to instantiate a font that cannot be located.

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

Constructors

FontNotAvailableException()

```
public FontNotAvailableException()
```

Constructs a `FontNotAvailableException` with `null` as its error detail message.

FontNotAvailableException(String)

```
public FontNotAvailableException(java.lang.String s)
```

Constructs a `FontNotAvailableException` with the specified detail message. The error message string `s` can later be retrieved by the `java.lang.Throwable.getMessage()` method of class `java.lang.Throwable`.

Parameters:

`s` - the detail message.

org.dvb.ui

TestOpacity

Declaration

```
public interface TestOpacity
```

All Known Implementing Classes:

```
org.havi.ui.HComponent
```

Description

Interface implemented by Components or Containers in order to allow the platform to query whether their paint method is fully opaque.

Methods

isOpaque()

```
public boolean isOpaque()
```

Returns true if the entire area of the component as given by the getBounds method, is fully opaque. Hence its paint method (or surrogate methods) guarantees that all pixels are painted in an opaque Color.

Classes implementing this interface shall return true from their implementation of this method if and only if their implementation can guarantee full opacity. The consequences of an invalid overridden value are implementation specific.

Returns:

true if all the pixels with the java.awt.Component#getBounds method are fully opaque, otherwise false.

org.dvb.ui

TextOverflowListener

Declaration

```
public interface TextOverflowListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```


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

Description

The TextOverflowListener is an interface that an application may implement and register in the DVBTextLayout-Manager. This listener will be notified if the text string does not fit within the component as a result of a call to the render method. It is the rendering process which triggers this event to be dispatched. The timing of this is implementation dependent.

Methods

notifyTextOverflow(String, HVisible, boolean, boolean)

```
public void notifyTextOverflow(java.lang.String markedUpString, org.havi.ui.HVisible v,
boolean overflowedHorizontally, boolean overflowedVertically)
```

This method is called by the DVBTextLayoutManager if the text does not fit within the component

Parameters:

`markedUpString` - the string that was successfully rendered within the component.

`v` - the HVisible object that was being rendered.

`overflowedHorizontally` - true if the text overflowed the bounds of the component in the horizontal direction; otherwise false.

`overflowedVertically` - true if the text overflowed the bounds of the component in the vertical direction; otherwise false.

org.dvb.ui

UnsupportedDrawingOperationException

Declaration

```
public class UnsupportedDrawingOperationException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.ui.UnsupportedDrawingOperationException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

The `UnsupportedDrawingOperationException` class represents an exception that is thrown if an drawing operation is not supported on this platform. E.g. DVBGraphics.setComposite could throw an Exception when setting the DST_IN rule on some devices while the SRC_OVER rule will always work.

Since:

MHP 1.0.

Constructors

UnsupportedDrawingOperationException(String)

```
public UnsupportedDrawingOperationException(java.lang.String s)
```

Constructs an instance of `UnsupportedDrawingOperationException` with the specified detail message.

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

Parameters:

s - the detail message.

Since:

MHP1.0.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 632 of 861

Annex V:
Void

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

Annex W (informative): DVB-J examples

W.1 DVB-J examples from MHP

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

W.2 Example of enumeration extension

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

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

public class AppRCTester {

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

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

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

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

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

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

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

public interface CATVRCInterface {

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

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

}
```

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

W.3 Example of testing for optional APIs

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

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

        ...

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

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

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

    /**
```

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

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

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

}


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

public class RCFeaturesImpl implements RCFeatures {

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

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

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


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

W.4 Example of lightweight trigger API

```
public class example implements StreamEventListener {

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

    ServiceDomain s = ....

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

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

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

W.5 Example of media stream synchronization API

```
import java.io.IOException;
import javax.media.Controller;
import javax.media.Manager;
import javax.media.NoPlayerException;
import javax.media.Player;
import javax.media.Time;
import javax.tv.service.selection.InsufficientResourcesException;

import org.davic.media.MediaLocator;
import org.davic.net.InvalidLocatorException;
import org.davic.net.dvb.DvbLocator;
import org.dvb.media.IncompatibleSynchronizableMediaTypeException;
import org.dvb.media.MasterSlaveSyncLinkageControl;
import org.dvb.media.SyncControl;
import org.dvb.media.SyncStateChangedEvent;
import org.dvb.media.SyncStateChangedEventListener;
import org.havi.ui.HPermissionDeniedException;
import org.havi.ui.HScreen;
import org.havi.ui.HVideoDevice;


public class SyncControlExample implements SyncStateChangedEventListener {

     private Player broadcastPlayer = null;
     private Player CCPlayer = null;
     private org.davic.media.MediaLocator CCMediaLocator;
     private DvbLocator CCLocator;
     private SyncControl bpSC = null;
     private SyncControl cpSC = null;
     private MasterSlaveSyncLinkageControl aMSSLC = null;

     public SyncControlExample() {

      // Create Broadcast Player
      HScreen screen = HScreen.getDefaultHScreen();
      HVideoDevice vd = screen.getHVideoDevices()[0];

      try {
         broadcastPlayer = (Player)vd.getVideoController();
      } catch (SecurityException e) {
         e.printStackTrace();
      } catch (HPermissionDeniedException e) {
         e.printStackTrace();
      }
      // Obtain SyncControl for Broadcast Player to be a slave Player
      bpSC = (SyncControl)broadcastPlayer.getControl("org.dvb.media.SyncControl");

      // Obtain MasterSlaveSyncLinkageControl to be a master Player
      aMSSLC =
```


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

```
(MasterSlaveSyncLinkageControl)broadcastPlayer.getControl("org.dvb.media.MasterSlaveSyncLinkageContr
ol");

      // Add listener for event related to synchronization
      aMSSLC.addSyncStateChangedEventListener(this);

      // Instantiate closed caption Player
      try {
         CCLocator = new DvbLocator("http://service/CCcomponent");
      } catch (InvalidLocatorException e2) {
         e2.printStackTrace();
      }
      CCMediaLocator = new MediaLocator(CCLocator);
      try {
         CCPlayer = Manager.createPlayer(CCMediaLocator);
      } catch (NoPlayerException e) {
         e.printStackTrace();
      } catch (IOException e) {
         e.printStackTrace();
      }

      // Obtain SyncControl to be a slave Player
      cpSC = (SyncControl)CCPlayer.getControl("org.dvb.media.SyncControl");

      if (CCPlayer != null) CCPlayer.start();

      Time tolerance = new Time(0.03); // Tolerance = 30ms
      Time offset = new Time(3.3); // Offset = 3.3s

      cpSC.setTolerance(tolerance);

      // Add closed caption player as a slave Player
      // Synchronization process starts automatically.
      try {
         aMSSLC.addSlave(CCPlayer, offset);
      } catch (InsufficientResourcesException e) {
         e.printStackTrace();
      } catch (IllegalArgumentException e) {
         // Checking reasons
         // Followings are sample to check whether CCPlayer is already a slave Player of other
master
         if (cpSC.getMaster() != null) {
            aMSSLC.removeSlave(CCPlayer);
         }
         // Followings are sample to check whether CCPlayer is a master
         // Instance of SyncControl can be created at most one.
         if
(((MasterSlaveSyncLinkageControl)CCPlayer.getControl("org.dvb.media.MasterSlaveSyncLinkageControl"))
.getSlaves().length != 0) {
            aMSSLC.removeSlave(CCPlayer);
         }
         e.printStackTrace();
      } catch (IncompatibleSynchronizableMediaTypeException e) {
         e.printStackTrace();
      } catch (IncompatibleTimeBaseException e) {
         e.printStackTrace();
      }

     }

     /*
      * Methods as an SyncStateChangedEvent event listener
      */

     public void inSync(SyncStateChangedEvent e) {

     }

     public void outOfSync(SyncStateChangedEvent e) {
      System.out.println("Now Resyncing...");
      Controller ctrl = e.getSourceController();
      int status = e.getSyncStatus();
      switch(status) {
      case MasterSlaveSyncLinkageControl.OUT_OF_SYNC_MASTER_CLK : {
         do_something;
         break;
      }
      case MasterSlaveSyncLinkageControl.OUT_OF_SYNC_SLAVE_CLK : {
         do_another_thing;
         break;
      }
       ....
      }

     }

     public void unSynchronizable(SyncStateChangedEvent e) {
      MasterSlaveSyncLinkageControl aSC = (MasterSlaveSyncLinkageControl)e.getSource();
      Controller aCtrl = e.getSourceController();
```


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

```
      aSC.removeSlave(aCtrl);  // Remove a slave Player that 'Give-up' to synchronize
     }
}
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 639 of 861

Annex X (normative): Test support

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

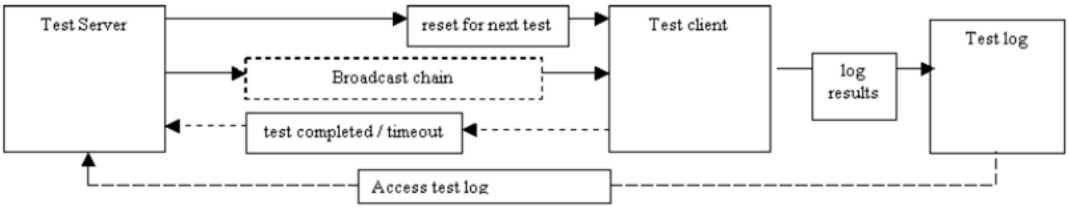
Package

Broadcast model

Description

In a broadcast-based conformance system, there are effectively three main entities involved in an automated test process:

1. The test server that is used to hold and initiate all of the tests.
2. The test client which runs the tests and logs the results.
3. The broadcast chain that is used to transfer applications and application data from the server to the client.



The communication order is as follows:

1. The test-server uses the "reset for next test" mechanism to set the test client into a known default state, ready to receive the test-application.
2. The test-server uses the "broadcast chain" mechanism to supply the test-application to the test client and to signal that the test-application should be executed.
3. The test-client runs the test-application.
4. The test-application either:
   - Finishes within a given timelimit, the result of the test is known and shall be considered to be the value reported by the test application for the purposes of compliance.
   - Optionally, the test-client may signal to the test-server that the test-application has finished executing and that the test-client is ready to be reset in order to receive the next test-application.
   - Fails to finish the test-application within a given timeout, the result of the test is unknown and shall be treated as a failure for the purposes of compliance. The test-server may treat the test-client as ready to be reset in order to receive the next application.

[Successive tests are then repeated from stage 1.]

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

"Reset for next test"

The "reset for next test" path is used by the test server to reset the test client to receive the next test. The reset for next test API is considered to be a private implementation issue between the test-server and test-client and therefore has no public Java API implications. Note that this "communication" needs to take place prior to any application being executed. Note that the precise manner of the reset mechanism is intentionally not specified - in the worst case, this may involve "power cycling" the test client.

Test log

Communication from the test client to the test log is considered as write-only access. Hence, results from successive tests cannot overwrite results from previous ones. Multiple (intermediate) results may be sent to the test log for any given test. It is recommended that all communication to the test log is synchronous.

See the DVBTest.log method for details of the proposed API and implementation issues.

"Test completed"

The "test completed" path is used by the test client to indicate to the test server that it has completed the previous test and is now able to accept a subsequent one. Note that this communication path is an optimization, since direct communication from the client to the server is not actually required, e.g. the server might simply "time-out" the client, and then perform a "reset for next test" action. However, this optimization may be important when large numbers of tests are being performed on a "capable" platform, since e.g. if a 30 second timeout is applied for 1 000 test cases which typically run within say 6 seconds, then the timeout implies a typical running time of 500 minutes, i.e. ~4,5 hours- rather than 100 minutes, i.e. ~1,5 hours.

See the DVBTest.terminate method for details of the proposed API and implementation issues.

Access test log

The mechanism by which the test-log is accessed is not considered in the present document, this is a private mechanism, which might include reading a file from flash/RAM. Similarly, the mechanism by which results are recovered from the test log is not considered in the present document, e.g. the test-log may actually reside on the test server, e.g. as in the case that results are transmitted over an IP connection.

Class Summary	
Classes	
DVBTest	The DVBTest class allows test applications to log messages during their execution and to indicate their termination condition in a platform independent manner.

org.dvb.test

DVBTest

Declaration

```
public class DVBTest

java.lang.Object
  |
  +--org.dvb.test.DVBTest
```

Description

The DVBTest class allows test applications to log messages during their execution and to indicate their termination condition in a platform independent manner.

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

A number of constants are defined in the DVBTest class and are reserved as follows:

- Zero and negative values defined within the class are reserved by DVB.
- Positive return values are available for test application specific return values, which must be defined within the procedure for executing the test application as to their precise meaning as regards conformance.

Fields

FAIL

```
public static final int FAIL
```

The application executed and terminated unsuccessfully and has therefore operated in a non-conformant manner.

HUMAN_INTERVENTION

```
public static final int HUMAN_INTERVENTION
```

The application is unable to determine whether it has operated conformantly and therefore requires some human intervention to determine whether conformance has been achieved. Until the application has been checked the result of the application should be considered as non-conformant.

It is envisaged that tests returning this value may be those requiring evaluation of presented content, such as graphics, etc. Such presentation may require (subjective) human evaluation.

OPTION_UNSUPPORTED

```
public static final int OPTION_UNSUPPORTED
```

The platform does not contain the option under test and therefore the test is inapplicable, the test result should not be considered when determining the status of the platform's conformance.

PASS

```
public static final int PASS
```

The application executed and terminated successfully and has therefore operated in a conformant manner.

UNRESOLVED

```
public static final int UNRESOLVED
```

A setup stage necessary to execute the application failed, and hence the result of the application is unknown and therefore should be considered to have operated in a non-conformant manner.

UNTESTED

```
public static final int UNTESTED
```

The application ran successfully, but the particular test was unable to execute. Hence the result of is unknown, and may require human evaluation to determine conformance.

For example, an out of disk space test may not execute within a fixed number of iterations (within a practical amount of time) for devices with large capacity storage, etc.

Methods

log(String, int)

```
public static void log(java.lang.String id, int no)
throws IOException
```

This method has the same behaviour, implementation options and restrictions as log(String, String) - except that. it allows an integer value to be logged, rather than a String, which may prove a useful option for automating tests.

Parameters:

`id` - a string identifying the application (thread) that is logging the test result.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 642 of 861

`no` - the integer value that the application wishes to be logged.

Throws:

`java.io.IOException` - under the same conditions as log(String, String).

log(String, String)

```
public static void log(java.lang.String id, java.lang.String message)
throws IOException
```

This synchronous, blocking, method logs a result (intermediate result) of a test application using write-only access. The method takes both an identifier string, e.g. "Test number 1" and a message to output, e.g. "Now invoking the xletPause method...". The application is not required to open a file or network connection, per se, and the log() method is always available for writing (in principle).

The precise format of the logged message is left deliberately unspecified, implementers may choose to output compressed messages, XML documents, or other formats of their choice (obviously provided that the original information can be recovered). It is an implementation option to include additional information with each logging message, e.g. including:

- version of the specification being implemented;
- compiler version and options;
- build-version;
- timestamp;
- date;
- debug info.

Messages sent using this method should "atomic", i.e. that they are not interleaved with other messages sent using the methods defined in the DVBTest class.

Implementation

The precise mechanism(s) by which the this method may be implemented are intentionally unspecified, implementation options might include:

- logging the message to a local file system;
- logging the message to a mounted remote file system;
- logging the message to a RAM disk, etc.;
- logging the message via an RS-232 (or other serial) connection;
- logging the message to a remote host via some IP / UDP based mechanism, e.g. using a socket-based connection.

Note that the implementation of the log method may use the same or a different mechanism to that used by the terminate method.

The log method does not require any explicit initialisation on the part of the application under test. For example if messages are being stored to a file system, then the application is not required to mount / open any storage file. Similarly, if the messages are being logged via a network connection, then the application is not required to open a connection to the storage host, etc. In principle, the mechanism should always be available to accept messages.

If this method is implemented on top of some buffering mechanism, it is strongly recommended that the buffer be flushed for each occurrence of a message being logged.

Security and implementation options

There is no Java security mechanism that is used to secure the log method.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 643 of 861

Note that even if the log method is based on a particular implementation option, it shall be able to operate in spite of that particular implementation option itself being subject to security checks. For example, a log method implemented using the java.net.Socket class shall always be able to log a message from a test-application, even if the test-application is unable to directly access the java.net.Socket class due to security restrictions, etc.

It is an allowed implementation option to require that the test-client be put into some particular "test-mode" before any test-results are logged. This mechanism is required to reduce any inadvertent interaction due to downloaded applications accessing the test methods.

Authoring guidelines

The log method is not intended to be accessed by downloaded applications directly, it is purely intended for the use of conformance test applications. Authors of downloaded applications should not call this method, since there may be interactions between this method and normal in-field operation of the test-client (GEM platform).

It is an allowed implementation option to require that the test-client be put into some particular "test-mode" before any test-results are logged. This mechanism is required to reduce any inadvertent interaction due to downloaded applications accessing the test methods.

It is an allowed implementation option to have a number of "test-modes" that are appropriate to different elements being conformance tested, for example, it is a valid implementation for a test-client to have a test-mode where results are stored via a serial port, and a separate test-mode where results are stored via a RAM disk. It is allowable for a conformance test to be performed with the test-client in some specific test- mode, e.g. a java.net test (using a serial modem) might have its test results logged to a RAM disk, to avoid interaction between test-log messages and the serial protocol.

The mechanism by which a test-client is put into a given test mode is intentionally left unspecified.

Relationship to java.io

It is an implementation option to map the implementation of this method onto corresponding write method(s) of appropriate java.io classes. These classes may in turn be obtained, e.g. from java.net Socket classes, etc.

Parameters:

`id` - a string identifying the application (thread) that is logging the test result.

`message` - the message that the application wishes to be logged.

Throws:

`java.io.IOException` - if there is any problem in providing synchronous logging to an application. This IOException may be due to failure to write to a file system, inability to access a remote socket, etc. the precise causes are deliberately unspecified and are implementation dependent.

prompt(String, int, String)

```
public static void prompt(java.lang.String id, int controlCode, java.lang.String message)
throws IOException
```

This is a method is used to "approximately" synchronise a test-client and test-server, the method blocks until the test-server positively or negatively acknowledges the particular message. The intended use of this method is to remove critical timing issues from conformance tests, e.g. a conformance test to ensure that an Xlet responds to a change in broadcast signalling must first ensure that the Xlet is in a state where it is able to respond to such signalling - since the time taken for an Xlet to achieve such a state is reliant on aspects outside of the scope of the conformance test itself (delivery bit rate, hardware and CPU capabilities of the test-client, etc.).

Messages sent using this method should "atomic", i.e. that they are not interleaved with other messages sent using the methods defined in the DVBTest class.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 644 of 861

Implementation

The precise mechanism(s) by which the this method may be implemented are intentionally unspecified. Implementation options for sending the prompt might include:

- logging the controlCode via an RS-232 (or other serial) connection;
- logging the controlCode to a remote host via some IP / UDP based mechanism, e.g. using a socket-based connection;
- displaying the message on-screen for a (human) test operator, e.g. for systems not implementing a return channel capability.

Implementation options for receiving the acknowledgement might include:

- acknowledgement via an RS-232 (or other serial) connection;
- acknowledgement from a remote host via some IP / UDP based mechanism, e.g. using a socket-based connection;
- a (human) test operator manually acknowledging the message, e.g. for systems not implementing a return channel capability.

Parameters:

`id` - a string identifying the application (thread) that is sending the prompt.

`controlCode` - an integer value (unique within a given Xlet) intended for use by some automated test process (corresponding to the readable message).

`message` - a message (unique within a given Xlet) intended to be readable by a (human) test operator (corresponding to the automated controlCode).

Throws:

`java.io.IOException` - If there is any problem in receiving a positive acknowledgement from the test-server, then an this shall be thrown. This may be due to a negative acknowledgement from the test-server, or due to other communication based causes - which are deliberately left unspecified.

terminate(String, int)

```
public static void terminate(java.lang.String id, int terminationCondition)
throws IOException
```

This synchronous, blocking, method logs the termination condition of a test application using write-only access. The method takes both an identifier string, e.g. "Test number 1" and a integer value to output, e.g. org.dvb.test.DVBTest.PASS. In addition to logging the termination condition of the test, invoking this method also indicates that the test application has terminated its operation. Note that termination of operation does not necessarily correspond to the application being in any particular lifecycle state (as defined in the "Application Model" clause of the GEM specification). The application is not required to open a file or network connection, per se, and the terminate() method is always available for writing (in principle).

The precise format of the termination message is left deliberately unspecified, implementers may choose to output compressed messages, XML documents, or other formats of their choice (obviously provided that the original information can be recovered). It is an implementation option to include additional information with each termination message, e.g. including:

- version of the specification being implemented;
- compiler version and options;
- build-version;
- timestamp;
- date;

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 645 of 861

- debug info.

On test-clients whose implementation of the terminate() method supports external communication to its test-server, implementations of this method may optionally indicate to the test-server that the test-client can be reset by its test-server so that another test may be initiated. The precise mechanism by which this communication takes place is not specified it may be via a IP / socket, serial port, etc.

In the case of an test-client that does not support communication to its test-server, or in the case of an unsuccessful (hanging) test, or inability of this method to return (without throwing an exception) the test-server must be prepared to "time out" the application running on the test-client and then reset the test-client.

Implementation

The precise mechanism(s) by which the this method may be implemented are intentionally unspecified, implementation options might include:

- storing the termination condition to a local file system;
- storing the termination condition to a mounted remote file system;
- storing the termination condition to a RAM disk, etc.;
- storing the termination condition via an RS-232 (or other serial) connection;
- storing the termination condition to a remote host via some IP / UDP based mechanism, e.g. using a socket-based connection.

Messages sent using this method should "atomic", i.e. that they are not interleaved with other messages sent using the methods defined in the DVBTest class.

Note that the implementation of the terminate method may use the same or a different mechanism to that used by the log method.

The terminate method does not require any explicit initialisation on the part of the application under test. For example if termination conditions are being stored to a file system, then the application is not required to mount / open any storage file. Similarly, if the results are being logged via a network connection, then the application is not required to open a connection to the storage host, etc. In principle, the mechanism should always be available to accept termination messages.

If this method is implemented on top of some buffering mechanism, it is strongly recommended that the buffer be flushed for each occurrence of a message being logged.

Security and implementation options

There is no Java security mechanism that is used to secure the terminate method.

Note that even if the terminate methods is based on a particular implementation option, it shall be able to operate in spite of that particular implementation option itself being subject to security checks. For example, a terminate method implemented using the java.net.Socket class shall always be able to log the termination condition of a test-application, even if the test-application is unable to directly access the java.net.Socket class due to security restrictions, etc.

It is an allowed implementation option to require that the test-client be put into some particular "test-mode" before any test-results are logged. This mechanism is required to reduce any inadvertent interaction due to downloaded applications accessing the test methods.

Authoring guidelines

The terminate method is not intended to be accessed by downloaded applications directly, it is purely intended for the use of conformance test applications. Authors of downloaded applications should not call this method, since there may be interactions between this method and normal in-field operation of the test-client (GEM platform).

It is an allowed implementation option to require that the test-client be put into some particular "test-mode" before any test-results are logged. This mechanism is required to reduce any inadvertent interaction due to downloaded applications accessing the test methods.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 646 of 861

It is an allowed implementation option to have a number of "test-modes" that are appropriate to different elements being conformance tested, for example, it is a valid implementation for a test-client to have a test-mode where results are stored via a serial port, and a separate test-mode where results are stored via a RAM disk. It is allowable for a conformance test to be performed with the test-client in some specific test- mode, e.g. a java.net test (using a serial modem) might have its test results logged to a RAM disk, to avoid interaction between test-log messages and the serial protocol.

The mechanism by which a test-client is put into a given test mode is intentionally left unspecified.

Relationship to java.io

It is an implementation option to map the implementation of this method onto corresponding write method(s) of appropriate java.io classes. These classes may in turn be obtained, e.g. from java.net Socket classes, etc.

Parameters:

`id` - a string identifying the application (thread) that is terminating the test.

`terminationCondition` - the termination condition of the test application.

Throws:

`java.io.IOException` - thrown if there is any problem in terminating an application. This may be due to failure to write to a file system, inability to access a remote socket, etc. the precise causes are deliberately unspecified.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 647 of 861

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

GEM accomplishes inter-application and inter-Xlet communication by means of the org.dvb.io.ixc API with the following modifications.

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

NOTE 1: organisation ID values between 0x80000000 and 0xffffffff are known to create problems in some implementations and their use is strongly discouraged.

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

NOTE 2: organisation ID values between 0x80000000 and 0xffffffff are known to create problems in some implementations and their use is strongly discouraged.

Package

org.dvb.io.ixc

Description

Provides support for inter-application communication.

Class Summary	
Classes	
IxcRegistry	This is the bootstrap mechanism for obtaining references to remote objects residing in other Xlets executing on the same GEM terminal, using a URL-like syntax.

org.dvb.io.ixc

IxcRegistry

Declaration

```
public class IxcRegistry

java.lang.Object
  |
  +--org.dvb.io.ixc.IxcRegistry
```

Description

This is the bootstrap mechanism for obtaining references to remote objects residing in other Xlets executing on the same GEM terminal, using a URL-like syntax. The identification of a remote object is given using a syntax indicating the organisation ID and application ID:

/organisation_id/application_id/name
organisation_id = the organisation ID of the Xlet, as signalled in the application_identifier record, defined in the GEM specification.
application_id = the application ID of the Xlet, as signalled in the application_identifier record, defined in the GEM specification.
name = the name under which the remote object was exported.

The organisation ID and the application ID shall each be encoded as a hexadecimal string, as would be accepted by java.lang.Integer.parseInt(String s, 16).

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

When RMI is used to communicate over a network, stubs generated by a tool like rmic are often required. This is not necessary for inter-xlet communication initiated with IxcRegistry. If such stubs are present, they shall be ignored.

Similarly, network RMI objects often extend the class server.RemoteObject, in order to get appropriate implementations for Object.hashCode(), Object.equals(), and Object.toString(). Overriding Object's implementation of these methods in this way is not necessary for inter-xlet communication initiated with IxcRegistry, although it is not harmful. Note that the class server.RemoteObject is not required in all GEM profiles.

Fields

GLOBAL

```
public static final int GLOBAL
```

Definition of the scope for bind or rebind - exported object is visible to any Xlet running within the same GEM terminal subject to requirements of the security model.

Since:

MHP 1.1.2.

PAGE

```
public static final int PAGE
```

Definition of the scope for bind or rebind - exported object is only visible to Xlets within the same DVB-HTML application. Note that an embedded Xlet with a different app ID than its enclosing HTML page is still considered to be the same application as that which contains the enclosing page.

Since:

MHP 1.1.2.

SERVICE

```
public static final int SERVICE
```

Definition of the scope for bind or rebind - exported object is only visible to Xlets running within the same service context

Since:

MHP 1.1.2.

Methods

bind(XletContext, String, Remote)

```
public static void bind(javax.tv.xlet.XletContext xc, java.lang.String name, java.rmi.Remote obj)
throws AlreadyBoundException
```

Exports an object under a given name in the namespace of an Xlet. The name can be any valid non-null String. No hierarchical namespace exists, e.g. the names "foo" and "bar/../foo" are distinct. If the exporting xlet has been destroyed, this method may fail silently.

The object shall be made visible to other applications running in the same service context. A call to bind(xc, name, obj) is thus equivalent to a call to bind(xc, name, obj, SERVICE).

Parameters:

`xc` - The context of the Xlet exporting the object.

`name` - The name identifying the object.

`obj` - The object being exported.

Throws:

`java.rmi.AlreadyBoundException` - if this Xlet has previously exported an object under the given name.

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

java.lang.NullPointerException - if xc, name or obj is null.

bind(XletContext, String, Remote, int)

```
public static void bind(javax.tv.xlet.XletContext xc, java.lang.String name, java.rmi.Remote obj,
int scope)
throws AlreadyBoundException
```

Exports an object under a given name in the namespace of an Xlet. The name can be any valid non-null String. No hierarchical namespace exists, e.g. the names "foo" and "bar/../foo" are distinct. If the exporting xlet has been destroyed, this method may fail silently.

Parameters:

xc - The context of the Xlet exporting the object.

name - The name identifying the object.

obj - The object being exported.

scope - The scope to which the object is to be exported.

Throws:

java.rmi.AlreadyBoundException - if this Xlet has previously exported an object under the given name.

java.lang.NullPointerException - if xc, name or obj is null.

Since:

MHP 1.1.

list(XletContext)

```
public static java.lang.String[] list(javax.tv.xlet.XletContext xc)
```

Returns an array of string path objects available in the registry. The array contains a snapshot of the names present in the registry that the current Xlet would be allowed to import using IxcRegistry.lookup.

Parameters:

xc - The context of the current Xlet.

Returns:

A non-null array of strings containing a snapshot of the path names of all objects available to the caller in this registry.

See Also:

lookup(XletContext, String)

lookup(XletContext, String)

```
public static java.rmi.Remote lookup(javax.tv.xlet.XletContext xc, java.lang.String path)
throws NotBoundException, RemoteException
```

Returns a remote object previously exported by an Xlet that has not been destroyed. The identification of a remote object is given using a syntax indicating the organisation ID and application ID:

/organisation_id/application_id/name
organisation_id = the organisation ID of the Xlet, as signalled in the application_identifier record.
application_id = the application ID of the Xlet, as signalled in the application_identifier record.
name = the name under which the remote object was exported.

The organisation ID and the application ID shall each be encoded as a hexadecimal string, as would be accepted by java.lang.Integer.parseInt(String s, 16). If the caller is not authorized to import a given object due to the security policy, then this API will behave as though the object had not been exported, that is, a NotBoundException shall be thrown.

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

Parameters:

`xc` - The context of the current Xlet (that is, the Xlet importing the object).

`path` - A file pathname-like string identifying the Xlet and the name of the object to be imported.

Returns:

A remote object.

Throws:

`java.rmi.NotBoundException` - If the path is not currently bound.

`java.rmi.RemoteException` - If a remote stub class cannot be generated for the object being imported.

`java.lang.IllegalArgumentException` - If the path is not formatted in the syntax given above.

`java.lang.NullPointerException` - if path is null.

rebind(XletContext, String, Remote)

```
public static void rebind(javax.tv.xlet.XletContext xc, java.lang.String name, java.rmi.Remote obj)
```

Rebind the name to a new object in the context of an Xlet; replaces any existing binding. The name can be any valid non-null String. No hierarchical namespace exists, e.g. the names "foo" and "bar/../foo" are distinct. If the exporting xlet has been destroyed, this method may fail silently.

The object shall be made visible to other applications running in the same service context. A call to rebind(xc, name, obj) is thus equivalent to a call to rebind(xc, name, obj, SERVICE).

Parameters:

`xc` - The context of the Xlet that exported the object.

`name` - The name identifying the object.

`obj` - The object being exported.

Throws:

`java.lang.NullPointerException` - if xc, name or obj is null.

rebind(XletContext, String, Remote, int)

```
public static void rebind(javax.tv.xlet.XletContext xc, java.lang.String name, java.rmi.Remote obj,
int scope)
```

Rebind the name to a new object in the context of an Xlet; replaces any existing binding. The name can be any valid non-null String. No hierarchical namespace exists, e.g. the names "foo" and "bar/../foo" are distinct. If the exporting xlet has been destroyed, this method may fail silently.

Narrowing the scope of the binding (e.g. from GLOBAL to SERVICE) shall have the same effect as a call to unbind for any applications which had references to that object and which were in scope but which are now out of scope.

Parameters:

`xc` - The context of the Xlet that exported the object.

`name` - The name identifying the object.

`obj` - The object being exported.

`scope` - The scope to which the object is to be exported.

Throws:

`java.lang.NullPointerException` - if xc, name or obj is null.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 651 of 861

Since:

MHP 1.1.

unbind(XletContext, String)

```
public static void unbind(javax.tv.xlet.XletContext xc, java.lang.String name)
throws NotBoundException
```

Unbind the name.

Parameters:

`xc` - The context of the Xlet that exported the object to be unbound.

`name` - The name identifying the object.

Throws:

`java.rmi.NotBoundException` - if this is not currently any object exported by this Xlet under the given name.

`java.lang.NullPointerException` - if xc or name is null.

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

Annex Z (informative): Services, service contexts and applications in a GEM environment

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

Z.1 Introduction

GEM describes the concepts that link the various parts of a GEM execution environment so that it can display a complete GEM service, including media and applications. This is really an overview to the GEM application lifecycle model, but does include some additional information.

We assume some familiarity with GEM and the Java TV specification.

Z.2 Basic concepts

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. In this version of the specification, the service is the contents of a service, including audio/video streams, data streams and all the service information, applications and application signalling that is being broadcast. The current service will largely be responsible for determining what media and applications are presented to the user.

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. A DVB-J application can call the select method on a service context (represented by javax.tv.service.selection.ServiceContext) and the platform will stop presenting all of the content that makes up the current service being presented by that service context and start presenting the content that makes up the new service. In this case, "content" may include one or more applications.

A service context has some major differences from a DVB-J Xlet context. It is not necessary to have one service context for every possible service that can or will get presented - one service context is needed for every service that can be presented simultaneously, but that is all. Also, a service context is not destroyed when the service within it is stopped, unlike the Xlet context for a DVB-J application. The service context exists until an application or the platform explicitly destroys it. In normal operational mode, the built-in navigator or EPG for a GEM system will create one single service context when it starts and never destroy that. GEM applications will run in that service context.

Z.3 Presenting a service in GEM

From the GEM point of view, the content of a service can be one of two types - media or applications. Media is the simplest case and is described first.

Z.3.1 Presenting the media components of a service

If there are several different streams of media that may get presented (such as several different video streams) then the platform uses a variety of methods to tell which streams should be used. These include user preferences and platform defaults, but will also include using service information do determine which streams get presented to the user.

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

For a DVB-J application, all real-time media components sharing the same clock are presented by the same JMF player. These players are directly linked to the service context, and the service context can be queried to find out which JMF Players are linked with it. It is also possible for applications to create JMF Player objects directly without linkage to the applications service context.

Z.3.2 Presenting the application components of a service

Applications are handled in a slightly different way. The lifecycle of all applications in a GEM environment is controlled by an application manager, a software entity that forms part of the GEM runtime environment. It takes its instructions on which applications to start and stop from the user, but also from information that is included in the GEM broadcast (called an "Application Description", can logically be considered an extension to MPEG's PMT) and the free resources in the platform.

The information carried in the Application Description not only says which applications are available, but also provides some instructions to the application manager about whether an application should be started automatically or whether an application should be killed automatically. The application manager monitors this information for changes, and creates, starts or kills applications as appropriate.

Every DVB-J application executes within an Xlet context. This is a similar concept to the applet context that a Java applet executes in, and it provides the Xlet with a link to its environment, both for accessing system properties and for telling the environment that the Xlet has changed its own state.

Since every Xlet executes as part of a service, there is also a link between the Xlet and its associated service. This link is the class javax.tv.service.selection.ServiceContextFactory. Using methods on this class, an Xlet can lookup its service context from its xlet context. From the service context, an Xlet can discover which service it is current running as part of. It can also register for events to be told when the service being presented in its service context changes.

The application manager maintains a list of all the applications in a system, so that it knows which ones are currently executing. It also knows (directly or indirectly) which applications are associated with which service context, so that if the service being presented in that service context changes due to a new service being selected, it can kill the applications that are not signalled in the new service. When a service is selected in a service context, the following steps happen in approximately this order:

a) The platform examines the MPEG PMT for the service that has been selected (tuning first if necessary) and works out which media streams are to be presented.

b) Any media streams currently playing are stopped and any new media streams that need to be presented are started. Any JMF players presenting the old content are stopped and destroyed, and players for the new content are created and started if necessary.

c) The platform monitors the application information table for the new service to find out which applications should be running. Any applications that are currently running but which are not signalled in the application information table of the new service (or which are signalled as killed) are killed and the Xlet contexts of DVB-J applications associated with the old service are destroyed.

d) For any applications which are signalled as autostart and are not currently running, the following steps are taken:

- The platform attaches to the object carousel signalled in the application information table.
- The platform attempts to load the main application file as signalled in the application information table.
- For a DVB-J application, the platform creates an instance of the main class using the default constructor, creates an XletContext object for the application and calls the Xlet.initXlet() method on the newly loaded class, passing the XletContext object as a parameter. Once this call is complete, the platform calls the Xlet.startXlet() method.

If several applications are signalled as autostart, the platform will load and start every one in the same way. Each DVB-J application will have a different Xlet context and will execute independently, although they are all associated with the same service context.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 654 of 861

Z.4 Service Context Object Design

Z.4.1 Overview

This clause explains the underlying design behind service contexts, resources such as video and audio decoders, and JMF players.

Z.4.2 Single-Application Model

For simplicity, we will first consider the operation of this API when a single application is executing.

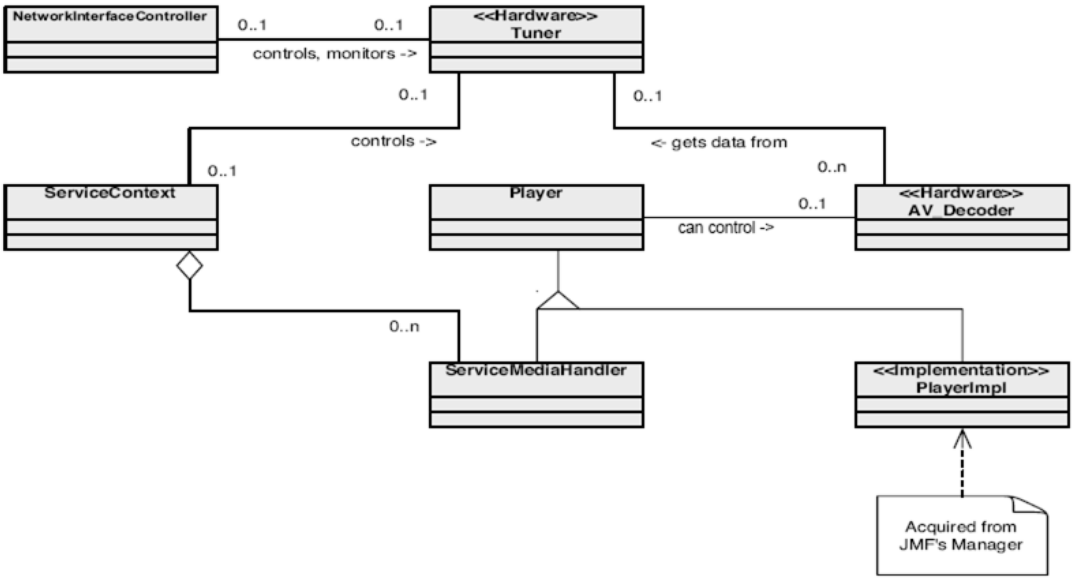
A service context represents the context in which services execute. There are different kinds of services, including application-only services, stand-alone stored services, services including audio/video content, and recorded services. All services may have applications, but some kinds of services cannot include audio/video content. Some, but not all services are played from a tuner. Thus, a service context:

- Has the ability to present applications.
- May control resources needed to present broadcast or recorded services, such as a tuner, a video decoder, and audio decoder.
- Is the entity an application accesses to select a new service.

Applications may select a different audio/video stream without selecting a new service. For example, an EPG that runs within an EPG service might wish to display the picture and play the audio associated with a different service, without actually selecting that service. This is not done by manipulating a service context; rather, it is done by using a JMF Player (which is not a `ServiceMediaHandler`), and if needed the tuning API. This is explained in greater detail below.

A service context is exposed to applications using the `javax.tv.service.ServiceContext` interface. The model underlying the `ServiceContext` interface is pictured in the following UML class diagram. The types shown without stereotypes indicating otherwise (e.g. "<<Hardware>>") are part of the public API. They are taken from different packages, but are pictured without the full package name for readability.

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



All `ServiceContext` instances act as a context to present services. Services may include applications, audio content, or video content. When a service containing A/V content is selected in a `ServiceContext`, the appropriate `ServiceMediaHandler` will attempt to acquire the underlying A/V decoder in hardware. If an application wishes to present a different A/V stream without selecting a new service (and thus, without potentially selecting a new set of applications), it cannot do so using the `ServiceMediaHandler` associated with the `ServiceContext`, because `ServiceMediaHandler` is restricted to only presenting streams within the service. Instead, the application may acquire a `Player` from the JMF `Manager`. This JMF player can be set to present a different A/V stream, but it is not allowed to tune. If that is required, the application acquires a DAVIC `NetworkController` to manually tune; on a GEM terminal with a single tuner, this would necessarily cause the underlying hardware for the tuner to be taken away from the `ServiceContext`.

Z.4.3 Multiple-Application Model

When multiple applications are running as part of the same service, it is important to understand the relationship between the `javax.tv.service.ServiceContext` instances in the two applications. Both applications may obtain `ServiceContext` instances using `ServiceContext.getServiceContext(XletContext)`, in order to manipulate the service context in which the two applications are running. In this case the two applications will, of course, have different instances of `ServiceContext`, but they represent the same underlying service context. That is, when one application uses its `ServiceContext` instance, for example to select a different service, it causes this action to happen in the underlying service context. If, then, the other application selects yet another service, this also occurs in the same underlying service context, so no additional acquisition of resources (e.g. as reported through the DAVIC resource management API) need take place.

Z.4.4 Considerations for standalone stored services

A standalone stored service contains only applications, and no A/V content. Further, because it is stored, no tuner is required. Thus, when a service context is presenting a standalone stored service, it is not required to be bound to a tuner or an audio or video decoder. However, the same service context may in the future be used to select a service that does include audio/video context, or that is a broadcast service received over a tuner. If such a service is selected, it is currently unspecified whether the tuner and A/V decoders remain bound to the service context during the presentation of the standalone stored service, or whether they are released and re-acquired. If an application has registered for `ResourceStatusEvent` notifications, it may or may not receive them, depending on the underlying implementation choice.

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

Z.5 Multiple service contexts in a GEM platform

The GEM specification allows the platform to have any number of service contexts, although the platform may choose to limit the number it can produce, possibly even to one. Each service context can present a different service, completely independently of the other service contexts. Operations carried out on one service context will not affect another (unless they cause tuning which prevents the contents of a service in another service context from being presented). It is even possible to display the same service in several service contexts simultaneously - this may not be very useful, but it is allowed. This may result in several instances of the same application running in at the same time in different service contexts. In practice, this is most likely to happen in future GEM terminals with multiple independent video output channels.

Z.6 How does the platform know which services are available?

In order to select a new service, an application (or the platform) has to know three things about the service it wants to start: the original network ID, the transport stream ID and the service ID for the service in question. In the case of an application, those values may be hard-wired into the application by the developer, but this not required. The application can find out about a service in the same way that the platform can - using service information.

The platform can not know the details of every available service when it started for the first time, and so it needs another method to find out about what services it can receive. A set-top box may do the following to find this out, for instance:

a) Scan the input (satellite, cable or some other input) to find out which transport streams are available and the physical parameters it needs to tune to them successfully.

b) For every transport stream:

- Tune to it.
- Use the service information API to access the service description table (SDT) and network information table (NIT) for that transport stream.
- Use these tables to find what services are in the transport stream and the values needed to select those services.
- In the case of the platform performing an initial scan, write this information to non-volatile memory.

c) When the application has found the service it wants (or in the case of the platform performing a scan, once every transport stream has been scanned in this way), a service can be selected.

At this point, the platform has all the information it needs to start presenting services to the user. It is important to realize that this is not the only way of finding this information - other ways may be possible, depending on the GEM implementation.

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

Annex AA:
Void

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

Annex AB:
Void

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

Annex AC:
Void

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

Annex AD:
Void

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

Annex AE (normative): Inner Applications

Package org.dvb.application.inner

Description

Provides support for embedding other interactive applications within the user interface of a DVB-J application.

Class Summary	
Interfaces	
DVBScene	Represents various features which are common to the top level Container of applications whether running stand-alone or embedded in a DVB-HTML application.
InnerApplicationListener	Listener for events related to inner applications
Classes	
InnerApplication	Encapsulates the information needed to define an inner application.
InnerApplicationContainer	Represents embedding of an inner application within the user interface of a DVB-J application.
InnerApplicationEvent	InnerApplicationEvent is used to notify applications which have registered interest in events coming from inner applications that such an event has happened.

org.dvb.application.inner DVBScene

Declaration

```
public interface DVBScene
```

Description

Represents various features which are common to the top level Container of applications whether running stand-alone or embedded in a DVB-HTML application. All integer constants used in this class are those defined in the class org.havi.ui.HScene.

Methods

addShortcut(int, HActionable)

```
public boolean addShortcut(int keyCode, org.havi.ui.HActionable comp)
```

Adds a shortcut key to action the specified `HActionable` . Generating the defined java.awt.KeyEvent or `HRcEvent` keycode causes the specified component to become actioned ——- potentially any keyCode may have a shortcut associated with it. The shortcut will only be added if the `HActionable` component is a child component in the container hierarchy of which the `HScene` is the root container. If this is not the case this method shall return false.

Note that a maximum of one `HActionable` may be associated with a given keyCode. An `HActionable` can have at most one associated shortcut keycode. Calling `addShortcut` repeatedly with the same `HActionable` will result in the previous short cut being removed. A short cut can be set on an invisible `HActionable` and therefore it is possible to provide short-cuts that have no user representation.

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

If the relevant keyCode is received by the HScene , then the HActionable will be actioned by the HScene sending it an ACTION_PERFORMED . Note that it is the responsibility of the application to ensure that the keyCode used is included in the set registered with the setKeyEvents method and is one which the platform can generate, i.e. where the method HRcCapabilities.isSupported() returns true.

Parameters:

keyCode - the keycode that represents the short cut. If keycode is java.awt.event.KeyEvent#VK_UNDEFINED, then the shortcut will not be added, any existing shortcut for this component will not be changed and this method shall return false.

comp - The actionable component that will be actioned.

Returns:

true if the shortcut was added, false otherwise.

addWindowListener(WindowListener)

```
public void addWindowListener(java.awt.event.WindowListener wl)
```

Add a listener to receive any java.awt.event.WindowEvents sent from this DVBScene . If the listener has already been added further calls will add further references to the listener, which will then receive multiple copies of a single event.

Parameters:

wl - The java.awt.event.WindowListener to be notified of any java.awt.event.WindowEvents.

enableShortcuts(boolean)

```
public void enableShortcuts(boolean enable)
```

Enables or disables all short cuts that are currently set on the Scene. To enable or disable a single shortcut use addShortcut or removeShortcut .

Note enableShortcuts(false) does not remove existing added DVBScene shortcuts - they are merely disabled and may be subsequently re-enabled with enableShortcuts(true) .

Parameters:

enable - a value of true indicates all shortcuts are to be enabled, and a value of false indicates all shortcuts are to be disabled.

getAllShortcutKeycodes()

```
public int[] getAllShortcutKeycodes()
```

Returns all keycodes added in the DVBScene as shortcuts.

Returns:

all keycodes added in the DVBScene as shortcuts, there are no ordering guarantees.

getBackgroundImage()

```
public java.awt.Image getBackgroundImage()
```

Retrieve any image used as a background for this DVBScene .

Returns:

an image used as a background, or null if no image is set. Note that depending on the current render mode any image set may not actually be rendered.

See Also:

setRenderMode(int)

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 663 of 861

getBackgroundMode()

```
public int getBackgroundMode()
```

Get the background mode of this DVBScene . The return value specifies whether the paint method should draw the background (i.e. a rectangle filling the bounds of the DVBScene).

Returns:

one of NO_BACKGROUND_FILL or BACKGROUND_FILL.

getFocusOwner()

```
public java.awt.Component getFocusOwner()
```

Returns the child component of this DVBScene which has focus if and only if this DVBScene is active.

Returns:

the component with focus, or null if no children have focus assigned to them.

getRenderMode()

```
public int getRenderMode()
```

Get the rendering mode of any background image associated with this DVBScene.

Returns:

the rendering mode, one of IMAGE_NONE , IMAGE_STRETCH , IMAGE_CENTER or IMAGE_TILE .

getShortcutComponent(int)

```
public org.havi.ui.HActionable getShortcutComponent(int keyCode)
```

Retrieve the HActionable associated with the specified shortcut key.

Parameters:

keyCode - the shortcut key code to be queried for an associated HActionable.

Returns:

the HActionable associated with the specified key if keyCode is a valid shortcut key for this DVBScene , null otherwise.

getShortcutKeycode(HActionable)

```
public int getShortcutKeycode(org.havi.ui.HActionable comp)
```

Returns the keycode associated with the specified HActionable component.

Parameters:

comp - the HActionable to return the keycode that it is associated with.

Returns:

the keycode associated with the specified HActionable component, if it is currently a valid shortcut "target", otherwise return java.awt.event.KeyEvent#VK_UNDEFINED.

isDoubleBuffered()

```
public boolean isDoubleBuffered()
```

Returns true if all the drawing done during the update and paint methods for this specific DVBScene object is automatically double buffered.

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

Returns:

`true` if all the drawing done during the update and paint methods for this specific `DVBScene` object is automatically double buffered, or `false` if drawing is not double buffered. The default value for the double buffering setting is platform-specific.

isEnableShortcuts()

```
public boolean isEnableShortcuts()
```

Returns the status of all short cuts that are currently set on the DVBScene.

Returns:

true if shortcuts are enabled, false otherwise.

See Also:

`enableShortcuts(boolean)`

isOpaque()

```
public boolean isOpaque()
```

Returns true if the entire `DVBScene` area, as given by the `java.awt.Component#getBounds` method, is fully opaque, i.e. its paint method (or surrogate methods) guarantee that all pixels are painted in an opaque `Color`.

By default, the return value depends on the value of the current background mode, as set by the `setBackgroundMode` method. The return value should be overridden by subclasses that can guarantee full opacity. The consequences of an invalid overridden value are implementation specific.

Returns:

`true` if all the pixels within the area given by the `java.awt.Component#getBounds` method are fully opaque, i.e. its paint method (or surrogate methods) guarantee that all pixels are painted in an opaque Color, otherwise `false`.

isVisible()

```
public boolean isVisible()
```

Determines if the `DVBScene` (or more properly its added child components) is Visible. Initially an `DVBScene` is invisible.

Returns:

true if the `DVBScene` is visible; false otherwise.

removeShortcut(int)

```
public void removeShortcut(int keyCode)
```

Removes the specified short-cut key. if the specified short-cut key is not registered, the method has no effect

Parameters:

`keyCode` - The keycode that represents the short cut.

removeWindowListener(WindowListener)

```
public void removeWindowListener(java.awt.event.WindowListener wl)
```

Remove a listener so that it no longer receives any java.awt.event.WindowEvents. If the specified listener is not registered, the method has no effect. If multiple references to a single listener have been registered it should be noted that this method will only remove one reference per call.

Parameters:

`wl` - The java.awt.event.WindowListener to be removed from notification of any java.awt.event.WindowEvents.

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

setBackgroundImage(Image)

```
public void setBackgroundImage(java.awt.Image image)
```

Set an image which shall be painted in the background of the DVBScene , after the background has been drawn according to the current mode set with setBackgroundMode, but before any children are drawn. The image is rendered according to the current render mode set with setRenderMode.

Note that the use of a background image in this way may affect the return value of the isOpaque method, depending on the image and the current rendering mode.

Parameters:

image - the image to be used as a background. If this parameter is null any current image is removed. Note that depending on the current render mode any image set may not actually be rendered.

See Also:

setRenderMode(int)

setBackgroundMode(int)

```
public void setBackgroundMode(int mode)
```

Set the background mode of this DVBScene . The value specifies whether the paint method should draw the background (i.e. a rectangle filling the bounds of the DVBScene).

Note that the background mode will affect the return value of the isOpaque method, depending on the value of the mode parameter. A fill mode of BACKGROUND_FILL implies that isOpaque must return true.

Parameters:

mode - one of NO_BACKGROUND_FILL or BACKGROUND_FILL . If mode is not a valid value, an IllegalArgument-Exception will be thrown.

setRenderMode(int)

```
public boolean setRenderMode(int mode)
```

Set the rendering mode of any background image associated with this DVBScene .

Note that the minimum requirement is to support only the IMAGE_NONE mode. Support of the other modes is platform and implementation specific.

Parameters:

mode - the rendering mode, one of IMAGE_NONE , IMAGE_STRETCH , IMAGE_CENTER or IMAGE_TILE .

Returns:

true if the mode was set successfully, false if the mode is not supported by the platform.

org.dvb.application.inner

InnerApplication

Declaration

```
public abstract class InnerApplication

java.lang.Object
  |
  +--org.dvb.application.inner.InnerApplication
```

Direct Known Subclasses:

```
org.dvb.dom.inner.HTMLApplication
```

Description

Encapsulates the information needed to define an inner application.

Constructors

InnerApplication()

```
protected InnerApplication()
```

A protected constructor. This shall not be used by inter-operable applications.

org.dvb.application.inner

InnerApplicationContainer

Declaration

```
public abstract class InnerApplicationContainer extends org.havi.ui.HComponent implements
org.havi.ui.HNavigable

java.lang.Object
  |
  +--java.awt.Component
        |
        +--org.havi.ui.HComponent
              |
              +--org.dvb.application.inner.InnerApplicationContainer
```

All Implemented Interfaces:

```
org.havi.ui.HMatteLayer, org.havi.ui.HNavigable, org.havi.ui.HNavigationInputPreferred,
java.awt.image.ImageObserver, java.io.Serializable, org.dvb.ui.TestOpacity
```

Direct Known Subclasses:

```
org.dvb.dom.inner.HTMLInnerApplicationContainer
```

Description

Represents embedding of an inner application within the user interface of a DVB-J application. The inner application shall be started when the object of this class is constructed. The behaviour of instances of this class when an inner application exits is either implementation or inner application format dependent.

When an instance of this class gains input focus,shall gain the input focus. The meaning of this is specific to the format of the inner application concerned. Inner applications are allowed to consume the events VK_UP, VK_DOWN, VK_LEFT and VK_RIGHT. While this is happening, the HNavigable focus management shall be disabled and the actions defined by `setFocusTraversal` shall not happen. The mechanisms by which inner applications start and stop consuming these events are specific to the content format of the inner application.

Constructors

InnerApplicationContainer(InnerApplication)

```
protected InnerApplicationContainer(org.dvb.application.inner.InnerApplication a)
throws IOException
```

Construct an instance of this class with a particular inner application as its content. This shall not be used by inter-operable applications.

Parameters:

`a` - the inner application.

Throws:

`java.io.IOException` - if an error occurred while reading the code or data for the inner application.

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

Methods

addHFocusListener(HFocusListener)

```
public void addHFocusListener(org.havi.ui.event.HFocusListener l)
```

Adds the specified HFocusListener to receive HFocusEvent events sent from this HNavigable. If the listener has already been added further calls will add further references to the listener, which will then receive multiple copies of a single event.

Specified By:

addHFocusListener in interface HNavigable

Parameters:

l - the HFocusListener to add.

addInnerApplicationListener(InnerApplicationListener)

```
public void addInnerApplicationListener(org.dvb.application.inner.InnerApplicationListener l)
```

Add a listener for events when the inner application exits.

Parameters:

l - the listener to be notified when the inner application exits.

getGainFocusSound()

```
public org.havi.ui.HSound getGainFocusSound()
```

Get the sound associated with the gain focus event.

Specified By:

getGainFocusSound in interface HNavigable

Returns:

The sound played when the component gains focus. If no sound is associated with gaining focus, then null shall be returned.

getLoseFocusSound()

```
public org.havi.ui.HSound getLoseFocusSound()
```

Get the sound associated with the lost focus event.

Specified By:

getLoseFocusSound in interface HNavigable

Returns:

The sound played when the component loses focus. If no sound is associated with losing focus, then null shall be returned.

getMove(int)

```
public org.havi.ui.HNavigable getMove(int keyCode)
```

Provides the HNavigable object that is navigated to when a particular key is pressed.

Specified By:

getMove in interface HNavigable

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 668 of 861

Parameters:

keyCode - The key code of the pressed key.

Returns:

Returns the HNavigable object, or if no HNavigable is associated with the keyCode then returns null.

getNavigationKeys()

```
public int[] getNavigationKeys()
```

Retrieve the set of key codes which this component maps to navigation targets.

Specified By:

getNavigationKeys in interface HNavigationInputPreferred

Returns:

an array of key codes, or null if no navigation targets are set on this component.

isSelected()

```
public boolean isSelected()
```

Indicates if this component has focus.

Specified By:

isSelected in interface HNavigable

Returns:

true if the component has focus, otherwise returns false.

processHFocusEvent(HFocusEvent)

```
public void processHFocusEvent(org.havi.ui.event.HFocusEvent evt)
```

Process an HFocusEvent sent to this HNavigationInputPreferred .

Specified By:

processHFocusEvent in interface HNavigationInputPreferred

Parameters:

evt - the HFocusEvent to process.

removeHFocusListener(HFocusListener)

```
public void removeHFocusListener(org.havi.ui.event.HFocusListener l)
```

Removes the specified HFocusListener so that it no longer receives HFocusEvent events from this HNavigable. If the specified listener is not registered, the method has no effect. If multiple references to a single listener have been registered it should be noted that this method will only remove one reference per call.

Specified By:

removeHFocusListener in interface HNavigable

Parameters:

l - the HFocusListener to remove.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 669 of 861

removeInnerApplicationListener(InnerApplicationListener)

```
public void removeInnerApplicationListener(org.dvb.application.inner.InnerApplicationListener l)
```

Remove a listener for events when the inner application exits.

Parameters:

`l` - the listener to be removed.

setFocusTraversal(HNavigable, HNavigable, HNavigable, HNavigable)

```
public void setFocusTraversal(org.havi.ui.HNavigable up, org.havi.ui.HNavigable down,
org.havi.ui.HNavigable left, org.havi.ui.HNavigable right)
```

Set the focus control for an `HNavigable` component. Note `setFocusTraversal` is a convenience function for application programmers where a standard up, down, left and right focus traversal between components is required.

Note `setFocusTraversal` is equivalent to multiple calls to `setMove` , where the key codes VK_UP, VK_DOWN, VK_LEFT, VK_RIGHT are used.

Note that this API does not prevent the creation of "isolated" `HNavigable` components —— authors should endeavor to avoid confusing the user.

Specified By:

`setFocusTraversal` in interface `HNavigable`

Parameters:

`up` - The `HNavigable` component to move to, when the user generates a VK_UP KeyEvent. If there is no `HNavigable` component to move "up" to, then null should be specified.

`down` - The `HNavigable` component to move to, when the user generates a VK_DOWN KeyEvent. If there is no `HNavigable` component to move "down" to, then null should be specified.

`left` - The `HNavigable` component to move to, when the user generates a VK_LEFT KeyEvent. If there is no `HNavigable` component to move "left" to, then null should be specified.

`right` - The `HNavigable` component to move to, when the user generates a VK_RIGHT KeyEvent. If there is no `HNavigable` component to move "right" to, then null should be specified.

setGainFocusSound(HSound)

```
public void setGainFocusSound(org.havi.ui.HSound sound)
```

Associate a sound with gaining focus, i.e. when the `HNavigable` receives a `java.awt.event.FocusEvent` event of type `FOCUS_GAINED`. This sound will start to be played when an object implementing this interface gains focus. It is not guaranteed to be played to completion. If the object implementing this interface loses focus before the audio completes playing, the audio will be truncated. Applications wishing to ensure the audio is always played to completion must implement special logic to slow down the focus transitions.

By default, an `HNavigable` object does not have any gain focus sound associated with it.

Note that the ordering of playing sounds is dependent on the order of the focus lost, gained events.

Specified By:

`setGainFocusSound` in interface `HNavigable`

Parameters:

`sound` - the sound to be played, when the component gains focus. If sound content is already set, the original content is replaced. To remove the sound specify a null `HSound` .

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 670 of 861

setLoseFocusSound(HSound)

```
public void setLoseFocusSound(org.havi.ui.HSound sound)
```

Associate a sound with losing focus, i.e. when the `HNavigable` receives a java.awt.event.FocusEvent event of type FOCUS_LOST. This sound will start to be played when an object implementing this interface loses focus. It is not guaranteed to be played to completion. It is implementation dependent whether and when this sound will be truncated by any gain focus sound played by the next object to gain focus.

By default, an `HNavigable` object does not have any lose focus sound associated with it.

Note that the ordering of playing sounds is dependent on the order of the focus lost, gained events.

Specified By:

`setLoseFocusSound` in interface `HNavigable`

Parameters:

`sound` - the sound to be played, when the component loses focus. If sound content is already set, the original content is replaced. To remove the sound specify a null `HSound` .

setMove(int, HNavigable)

```
public void setMove(int keyCode, org.havi.ui.HNavigable target)
```

Defines the navigation path from the current `HNavigable` to another `HNavigable` when a particular key is pressed.

Note that `setFocusTraversal` is equivalent to multiple calls to `setMove` , where the key codes `VK_UP`, `VK_DOWN`, `VK_LEFT`, `VK_RIGHT` are used.

Specified By:

`setMove` in interface `HNavigable`

Parameters:

`keyCode` - The key code of the pressed key. Any numerical keycode is allowed, but the platform may not be able to generate all keycodes. Application authors should only use keys for which `HRcCapabilities.isSupported()` or `HKeyCapabilities.isSupported()` returns true.

`target` - The target `HNavigable` object that should be navigated to. If a target is to be removed from a particular navigation path, then `null` should be specified.

org.dvb.application.inner

InnerApplicationEvent

Declaration

```
public class InnerApplicationEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.application.inner.InnerApplicationEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

InnerApplicationEvent is used to notify applications which have registered interest in events coming from inner applications that such an event has happened.

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

Constructors

InnerApplicationEvent(InnerApplicationContainer)

```
public InnerApplicationEvent(org.dvb.application.inner.InnerApplicationContainer source)
```

Constructor for an event.

Parameters:

`source` - the `InnerApplicationContainer` which generated the event.

org.dvb.application.inner

InnerApplicationListener

Declaration

```
public interface InnerApplicationListener
```

Description

Listener for events related to inner applications.

Methods

InnerApplicationExited(InnerApplicationEvent)

```
public void InnerApplicationExited(org.dvb.application.inner.InnerApplicationEvent e)
```

Called when an inner application exits.

Parameters:

`e` - an event object encapsulating what happened.

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

Annex AF (normative): Plug-in APIs

NOTE: Support for the classes `XletContainer` and `XletSystemCall` is optional in the present document.

Package org.dvb.application.plugins

Description

Provides support for inter-operable plug-ins in the GEM specification. The following classes and interfaces are intended for generic plug-ins and are not specific to any particular content format.

- InvalidApplicationException.
- Plugin.
- ApplicationSecurityContext.

The following classes are intended specifically for DVB-HTML plug-ins.

- XletContainer.
- XletSystemCall.

Class Summary	
Interfaces	
Plugin	This interface allows an application or service manager to control the lifecycle of applications being executed by an inter-operable plug-in.
Classes	
ApplicationSecurityContext	This class represents the security context for an application whose permissions are defined by the GEM security model, in particular a permission request file.
XletContainer	This class provides a container that can be embedded in a widget hierarchy, and can provide a container that can safely be used as the top-level container of an embedded Xlet.
XletSystemCall	This class permits user code to intercept certain system calls initiated by an embedded Xlet that need to be serviced by a support application.
Exceptions	
InvalidApplicationException	Thrown if an application is not valid for execution by a plug-in.

org.dvb.application.plugins ApplicationSecurityContext

Declaration

```
public class ApplicationSecurityContext

java.lang.Object
  |
  +--org.dvb.application.plugins.ApplicationSecurityContext
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 673 of 861

Description

This class represents the security context for an application whose permissions are defined by the GEM security model, in particular a permission request file. One example of this is DVB-HTML applications managed by a plug-in application. This would also apply to applications in other content format where the permissions for that content format are signalled with GEM mechanisms and governed by the GEM security model.

Constructors

ApplicationSecurityContext(URL[], URL, AppID)

```
public ApplicationSecurityContext(java.net.URL[] path, java.net.URL entryPoint,
org.dvb.application.AppID appID)
throws IOException
```

Creates a new security context for an application whose code can be found on the path supplied, and whose entry point directory is as given. Under no circumstances will an application managed by a plug-in be given access to resources not available to the plug-in itself. This policy is reflected in the permissions granted to the ApplicationSecurityContext, as well as the permissions granted to any classes loaded by any class loaders managed by a security context.

If there is a permission request file in the directory identified by the entryPoint, it will be processed in the same way as the permission request file of a DVB-J application. i.e. reading it in, parsing it and taking account of the access rights granted by the user as defined under "General principles" in the main body of the present document.

Parameters:

`path` - The search path for locating resources within the application.

`entryPoint` - The directory containing the permission request file to use.

`appID` - the application ID which the application is to run under.

Throws:

IOException - when there is an IO error reading in the permission request file or attempting to read in the permission request file or attempting to discover the existence of a permission request file.

`java.lang.NullPointerException` - if entryPoint is null, if path is null, or if any element of path is null.

`java.lang.IllegalArgumentException` - if path.length < 1.

`java.io.IOException`.

Methods

checkPermission(Permission)

```
public void checkPermission(java.security.Permission p)
```

Throws a `SecurityException` if the requested access, specified by the given permission, is not permitted to the application or sub-application represented by this application security context object. The set of permissions granted to an entity is be a function of receiver policy, possibly influenced by user settings, application signer, and permission request file.

Parameters:

`p` - A permission object representing the resource for which access is being checked.

Throws:

`java.lang.NullPointerException` - if p is null.

`java.lang.SecurityException` - if this application has not been granted access to the resource represented by p.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 674 of 861

createEmbeddedContext(URL[], URL)

```
public org.dvb.application.plugins.ApplicationSecurityContext createEmbeddedContext(java.net.URL[]
path, java.net.URL entryPoint)
```

Creates a context for an embedded part of an application, e.g. an Xlet embedded within a DVB-HTML page. The set of permissions granted to the application will be the same as the parent ApplicationSecurityContext.

Parameters:

`path` - The search path for locating resources within the application.

`entryPoint` - The resource of the entry point of this application.

Returns:

a context for the part of the application concerned.

Throws:

`java.lang.NullPointerException` - if entryPoint is null, if path is null, or if any element of path is null.

`java.lang.IllegalArgumentException` - if path.length < 1.

doPrivileged(PrivilegedAction)

```
public java.lang.Object doPrivileged(java.security.PrivilegedAction action)
```

Performs the specified PrivilegedAction with privileges enabled and restricted by the specified AccessControlContext. The action is performed with the intersection of the permissions possessed by the caller's protection domain, and those possessed by the domains represented by the specified AccessControlContext. If the action's run method throws an (unchecked) exception, it will propagate through this method.

Parameters:

`action` - the action to be performed.

Returns:

the value returned by the action's run method.

getClassLoader(String[])

```
public java.lang.ClassLoader getClassLoader(java.lang.String[] forbiddenPackages)
```

Get a classloader that is appropriate for loading classes for the application (or sub-application) represented by this application security context object. If this method is called more than once, the same instance will be returned.

It is important that embedded DVB-J code be prevented from accessing classes that implement the plug-in application. To this end, the plug-in may specify a list of forbidden packages. Classes loaded by the returned classloader will be forbidden from loading or accessing classes in the named packages.

Parameters:

`forbiddenPackages` - a list of forbidden package names, e.g. `{ "de.tu-bs.ing.ifn.plugin.impl" }`.

Returns:

A class loader that is appropriate for loading and DVB-J classes that are a part of this application or sub-application.

getResource(String, boolean)

```
public java.net.URL getResource(java.lang.String name, boolean sameSigner)
```

Get a locator to the named resource, within the search path for this application. Will return null if a resource with the given name that is appropriately signed (if necessary) cannot be found.

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

NOTE: This method can be used, for example, by an interoperable plug-in that needs to fetch part of an application that is not loaded by a classloader. For example, it could be used to get a locator to an HTML page, if and only if that page is appropriately signed.

Parameters:

`name` - The name of the resource (e.g. com/foo/MyPage.html).

`sameSigner` - True if this is code, or any other resource for which the signer must be the same as the signer of the entry point.

Returns:

URL to the named resource, or null.

org.dvb.application.plugins

InvalidApplicationException

Declaration

```
public class InvalidApplicationException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.application.plugins.InvalidApplicationException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown if an application is not valid for execution by a plug-in.

Since:

MHP1.1.

Constructors

InvalidApplicationException()

```
public InvalidApplicationException()
```

Construct a InvalidApplicationException with no detail message.

InvalidApplicationException(String)

```
public InvalidApplicationException(java.lang.String s)
```

Construct a InvalidApplicationException with a detail message.

Parameters:

`s` - detail message.

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

org.dvb.application.plugins

Plugin

Declaration

```
public interface Plugin
```

Description

This interface allows an application or service manager to control the lifecycle of applications being executed by an inter-operable plug-in. It shall be implemented by inter-operable plug-ins and shall be called by implementation of the application manager in a GEM terminal. It shall not be called by GEM applications. The usage model for this interface is the reuse of the `javax.tv.xlet.Xlet` interface to represent applications in formats other than DVB-J. It is the responsibility of the plug-in to provide implementations of the methods in that interface and to perform the translation between the semantics specified for DVB-J applications and the semantics of the application format which the plug-in supports. The plug-in is initialized by the application manager calling the no-argument constructor of the object implementing this interface. Plug-ins are not expected to perform any time consuming activities or hold any scarce resources in this call.

Methods

initApplication(AppAttributes)

```
public javax.tv.xlet.Xlet initApplication(org.dvb.application.AppAttributes app)
throws InvalidApplicationException
```

Request the plug-in to prepare for executing an instance of an application in a format which it supports. If the application cannot be prepared then null shall be returned. While the plug-in is in the terminated state then this method shall always return null. The application shall only be considered to be started when the application manager calls the methods on the Xlet interface returned by this method.

Parameters:

`app` - an instance of `AppAttributes` for the application which is to be started.

Returns:

an object implementing the Xlet interface or null if the application cannot be started.

Throws:

`InvalidApplicationException` - if the application to be started is not valid for this plug-in.

initApplication(InnerApplication)

```
public javax.tv.xlet.Xlet initApplication(org.dvb.application.inner.InnerApplication app)
throws InvalidApplicationException
```

Request the plug-in to prepare for executing an instance of an application in a format which it supports.If the application cannot be prepared then null shall be returned. While the plug-in is in the terminated state then this method shall always return null. The application shall only be considered to be started when the application manager calls the methods on the Xlet interface returned by this method.

Parameters:

`app` - an instance of InnerApplication for the application which is to be started.

Returns:

an object implementing the Xlet interface or null if the application cannot be started.

Throws:

`InvalidApplicationException` - if the application to be started is not valid for this plug-in.

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

initPlugin()

```
public boolean initPlugin()
```

Initialize the plug-in. Any time consuming initializations should be performed during this method call. This method shall be called after the no-argument constructor of the object implementing this interface and before any calls to the `startApplication`. It may also be called after a call to `terminatePlugin` if a plug-in was not removed from the virtual machine where it was executing. The behaviour of an application manager should a plug-in fail to initialize is intentionally unspecified.

Returns:

true if the plug-in initialized successfully otherwise false.

isSupported(AppAttributes)

```
public boolean isSupported(org.dvb.application.AppAttributes app)
```

Test whether this plug-in is able to support a particular application. This method shall work regardless of whether the plug-in has been initialized or not.

Parameters:

`app` - an instance of `AppAttributes` for the application which is to be started.

Returns:

true if the plug-in can support this application otherwise false.

terminatePlugin()

```
public void terminatePlugin()
```

Terminate the plug-in. This method shall only be called after all applications being executed by this plug-in have been terminated. The implementation of this method shall release all resources held by the plug-in. Once this method has returned, the plug-in shall either be unloaded from the virtual machine where it was executing or the `initPlugin` method shall be called again to put the plug-in again in an initialized condition. A plug-in may not refuse to terminate and throwing a runtime exception or error from this method shall result in the plug-in being removed from the virtual machine where it was executing.

org.dvb.application.plugins

XletContainer

Declaration

```
public class XletContainer extends java.awt.Container
```

```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--org.dvb.application.plugins.XletContainer
```

All Implemented Interfaces:

```
java.awt.image.ImageObserver, java.io.Serializable
```

Description

This class provides a container that can be embedded in a widget hierarchy, and can provide a container that can safely be used as the top-level container of an embedded Xlet. This solves the problem of an embedded Xlet being able to call getParent() to discover the widget hierarchy in which it is embedded.

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

Since:

MHP 1.1.

Constructors

XletContainer(Container)

```
public XletContainer(java.awt.Container parent)
```

Construct a new XletContainer as a child of Parent. The XletContainer contains exactly one widget: A child container. The child container is constrained to be at the same location and the same size as the parent XletContainer.

Parameters:

`parent` - The widget that is to have an Xlet embedded within it.

Throws:

`java.lang.NullPointerException` - if parent is null.

See Also:

`getXletContainer()`

Methods

getXletContainer()

```
public java.awt.Container getXletContainer()
```

Get the only child of the XletContainer. This container can safely be used as the top-level container of an Xlet, e.g. an Xlet embedded in a DVB-HTML page. Calling getParent() on this container from client code will return null, unless the caller has been granted special access via a platform-dependent mechanism.

If this method is invoked more than once for the same instance of XletContainer, it will return the same instance.

Returns:

The child of this XletContainer.

org.dvb.application.plugins

XletSystemCall

Declaration

```
public abstract class XletSystemCall
```

```
java.lang.Object
  |
  +--org.dvb.application.plugins.XletSystemCall
```

Description

This class permits user code to intercept certain system calls initiated by an embedded Xlet that need to be serviced by a support application. For example, a DVB-HTML plug-in application needs to service requests that are made by an embedded Xlet, typically via static method calls.

Since:

MHP1.1.

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

Constructors

XletSystemCall()

```
protected XletSystemCall()
```

Create a new XletSystemCall.

Methods

getRootContainer(XletContext)

```
public abstract java.awt.Container getRootContainer(javax.tv.xlet.XletContext ctx)
```

Called when the Xlet calls javax.tv.graphics.TVContainer.getRootContainer().

Parameters:

`ctx` - The context of the Xlet making the request; it shall be identical to the XletContext used to create this instance of XletSystemCall.

Returns:

a container object to be returned to the embedded xlet

See Also:

`javax.tv.graphics.TVContainer.getRootContainer(XletContext)`

register(Plugin, XletContext)

```
public final void register(org.dvb.application.plugins.Plugin p, javax.tv.xlet.XletContext ctx)
```

Register this instance of XletSystemCall with the system.

Parameters:

`p` - The Plugin that services calls made by the xlet, i.e. the Plugin of which this instance of XletSystemCall is a part.

`ctx` - The XletContext of the Xlet making the calls.

Throws:

`java.lang.NullPointerException` - if p or ctx is null.

See Also:

`unregister(Plugin, XletContext)`

unregister(Plugin, XletContext)

```
public final void unregister(org.dvb.application.plugins.Plugin p, javax.tv.xlet.XletContext ctx)
```

Unregister this instance of XletSystemCall with the system. When an interoperable Plugin terminates, of an Xlet managed by a Plugin is asked to terminate, the Plugin must unregister any relevant XletSystemCall instances.

Parameters:

`p` - The Plugin that services calls made by the xlet, i.e. the Plugin of which this instance of XletSystemCall is a part.

`ctx` - The XletContext of the Xlet making the calls.

Throws:

`java.lang.NullPointerException` - if p or ctx is null.

See Also:

`register(Plugin, XletContext)`

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

Annex AG (normative): Stored application APIs

Package org.dvb.application.storage

Description

Provides support for storage of applications.

Class Summary	
Interfaces	
ApplicationStorageController	Defines the methods for storing, listing and removing applications to and from a service.
ApplicationStorageListener	Listener to receive reports on progress of application storage operations.
ExtendedAppAttributes	The `ExtendedAppAttributes` interface provides additional attributes that are useful when application can be stored in the GEM terminal.
StoredApplicationService	Defines the information about a stored application service.
Classes	
ApplicationCache	A cache to store applications so that they may be started faster when signalled in broadcast.
ApplicationStoragePermission	This class represents a permission to manage applications stored in the GEM terminal.
StoredApplicationServiceFactory	This factory creates new Service objects representing stand-alone stored application services.
StoredApplicationServiceType	The service type used for stored application services.
Exceptions	
ApplicationDownloadException	Thrown if the downloading of the application failed.
InvalidApplicationException	When an application is being installed into a service, this is thrown when: The application does not include an `application_storage_descriptor`.
InvalidDescriptionFileException	Thrown when an application descriptio file is invalid.
NotEnoughResourcesException	Thrown if the GEM terminal does not have enough resources to install the application.
ServiceAlreadyExistsException	Thrown if a StoredApplicationService already exists with the identifiers the application tried to create a new one.
UserRejectedInstallException	Thrown if the request to install an application was rejected.

org.dvb.application.storage ApplicationCache

Declaration

```
public abstract class ApplicationCache

java.lang.Object
   |
   +--org.dvb.application.storage.ApplicationCache
```


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

Description

A cache to store applications so that they may be started faster when signalled in broadcast.

Applications stored via this mechanism cannot be started directly, only via an AIT entry in a broadcast service.

Note that applications are globally uniquely identified by a combination of organisation ID, application ID, and version number. Application authors should ensure that their storable applications are consistently broadcast.

Each instance of ApplicationCache can store only a single version of an application - i.e. within an `ApplicationCache`, an organisation ID and application ID pair can uniquely identify an application. Successfully storing a different version of a cached application than the current one in an ApplicationCache shall result in the current one being removed from that ApplicationCache instance. If the call to store fails then the current version shall not be removed.

A single GEM terminal will support many instances of `ApplicationCache`. These shall share a single underlying application store, and that underlying store shall use reference-counting or similar techniques to allow several stored services and `ApplicationCache` instances to contain the same application without wasting storage space by storing the same application multiple times. Space permitting, the underlying application store shall support storing several different versions of a single application (i.e. same organisation ID and application ID but different version numbers) where these are stored via different `ApplicationCache` instances or form part of different stored services. The underlying application store shall manage the removal of versions of stored applications which are not used in any stored services or application caches.

If an application is already cached in one `ApplicationCache`, storing the same version of the same application in a different `ApplicationCache` shall not fail due to insufficient disk space, and if all files in the ADF are labelled as critical, it shall not fail due to inability to download files listed in the ADF and shall complete quickly (without waiting for the files listed in the ADF to be broadcast).

Since:

MHP1.1.2.

Constructors

ApplicationCache()

```
protected ApplicationCache()
```

Interoperable applications shall not call this constructor.

Since:

MHP1.1.2.

Methods

getDefaultCache()

```
public static org.dvb.application.storage.ApplicationCache getDefaultCache()
```

Get the default cache that this application can use. Note that in MHP 1.1, applications cannot access any cache other than the default one.

The object returned depends on the organisation_id of the calling application. Applications with the same organisation_id shall recieve the same `ApplicationCache`. Applications with different organisation_id shall recieve different, independent instances of `ApplicationCache`. (I.e. if one application stores an application or removes a stored application, that operation shall be visible to another application via the methods on this class if and only if the other application has the same organisation_id as the first application).

Returns:

An `ApplicationCache` instance.

Throws:

`java.lang.SecurityException` - Thrown if the application calling this method does not have an `ApplicationStoragePermission` with action "manageCache" for it's own organisation_id.

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

Since:

MHP1.1.2.

getStoredAppIDs()

```
public abstract org.dvb.application.AppID[] getStoredAppIDs()
```

Lists the `AppID`s of the applications that are stored within this cache.

Returns:

an array of `AppID` objects representing the stored applications.

Since:

MHP1.1.2.

getVersionNumber(AppID)

```
public abstract int getVersionNumber(org.dvb.application.AppID appId)
```

Return the version number of the stored application whose `AppID` is given as a parameter.

Parameters:

`appId` - the `AppID` of the application whose version is queried.

Returns:

the version number of the stored application, returns -1 if the application given as a parameter is not stored as part of this cache.

Since:

MHP1.1.2.

remove(AppID)

```
public abstract void remove(org.dvb.application.AppID appId)
```

Requests the GEM terminal to initiate the removal of an application stored in the GEM terminal from this cache.

The platform is not expected to consult the end user of the GEM terminal for the permission to remove the application.

This specification does not prevent a terminal keeping an application in cache even after all references to it have been removed via this API.

If the application identified by the AppID passed in as a parameter is not installed in this cache, the method shall fail silently.

If the application is already stored elsewhere (e.g. in another ApplicationCache, or in a StoredApplicationService) then those caches and stored services shall not be affected by calling this method.

Parameters:

`appId` - `AppID` of the application to be removed.

Throws:

`java.lang.SecurityException` - thrown if the application calling this method does not have an `ApplicationStoragePermission` with action "removeCache" for the organisation_id of the application to be removed and an ApplicationStoragePermission with action "manageCache".

Since:

MHP1.1.2.

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

store(AppProxy, boolean)

```
public abstract void store(org.dvb.application.AppProxy app, boolean canPrompt)
throws InvalidApplicationException, NotEnoughResourcesException, InvalidDescriptionFileException, Ap
plicationDownloadException
```

Store an application in the GEM terminal.

GEM terminals may prompt the user for permission to free up resources if and only if all of the following conditions hold:

- `canPrompt` is true;
- the caller has the necessary permissions (i.e. the `SecurityException` will not be thrown);
- the application is valid (i.e. the `InvalidApplicationException` will not be thrown);
- the application description file is valid (i.e. the `InvalidDescriptionFileException` will not be thrown);
- the GEM terminal determines that there are insufficient resources to store the application, and it cannot or will not silently free up enough resources (i.e. the `NotEnoughResourcesException` would be thrown);
- the GEM terminal determines that it could free up sufficient resources to store the application (i.e. to ensure the `NotEnoughResourcesException` would not be thrown) if it had the user's consent;

Prompting the end-user for permission is not required however applications setting canPrompt to true should be prepared for the possibility of it happening. Note that if the user decides not to allow the terminal to free up resources, the NotEnoughResourcesException will be thrown (as if the terminal had not asked the user at all).

NOTE: This method is synchronous and will block until the storing is either completed or fails.

Storing (the same version of) an application that is already stored elsewhere (e.g. in another ApplicationCache, or in a StoredApplicationService) shall use the already-stored files.

Parameters:

`app` - an AppProxy representing the application to be installed.

`canPrompt` - If `true`, the terminal is allowed to prompt the user for permission to free up space in order to store this application. If `false`, the terminal shall not prompt the user for permission to cache this application.

Throws:

`InvalidApplicationException` - thrown if the specified application is not signalled as capable of being stored, or if the specified application is not signalled as part of the same service as the calling application.

`NotEnoughResourcesException` - thrown if the GEM terminal does not have enough resources, e.g. storage space, available for the application.

`InvalidDescriptionFileException` - thrown if the application description file is missing, invalid or otherwise not conformant to the specification.

`ApplicationDownloadException` - thrown if the downloading of the application files was not successful (e.g. a carousel error, a file in the application description file is missing in the carousel, if the application was removed from the AIT or from its transport mechanism while installation is in progress, etc).

`java.lang.SecurityException` - Thrown if the application calling this method does not have an `ApplicationStoragePermission` with action "storeCache" or "*" for the organisation_id of the application to be stored.

Since:

MHP1.1.2.

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

org.dvb.application.storage ApplicationDownloadException

Declaration

```
public class ApplicationDownloadException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.application.storage.ApplicationDownloadException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown if the downloading of the application failed. E.g. a carousel error, a file in the application description file is missing in the carousel, the application was being authenticated as part of downloading and this failed, the application was specified as being stored from another service and that service is not currently reachable (e.g. requires tuning), etc.

Since:

MHP1.1.

Constructors

ApplicationDownloadException()

```
public ApplicationDownloadException()
```

Constructs an `ApplicationDownloadException` with no detail message.

ApplicationDownloadException(String)

```
public ApplicationDownloadException(java.lang.String s)
```

Constructs an `ApplicationDownloadException` with a detail message

Parameters:

`s` - detail message.

org.dvb.application.storage ApplicationStorageController

Declaration

```
public interface ApplicationStorageController
```

All Known Subinterfaces:

`StoredApplicationService`, `org.dvb.service.UnboundApplicationService`

Description

Defines the methods for storing, listing and removing applications to and from a service.

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

Since:

MHP1.1.3.

Methods

getStoredAppIDs()

```
public org.dvb.application.AppID[] getStoredAppIDs()
```

Lists the AppIDs of the applications that are stored within this service.

Returns:

an array of AppID object representing the stored application.

Throws:

java.lang.SecurityException - Thrown if the application calling this method does not have an ApplicationStoragePermission with action "manageService" for the organisation_id of this service.

Since:

MHP1.1.2.

getVersionNumber(AppID)

```
public int getVersionNumber(org.dvb.application.AppID appId)
```

Return the version number of the stored application whose AppID is given as a parameter.

Parameters:

appId - the AppID of the application whose version is queried.

Returns:

the version number of the stored application, returns -1 if the application given as a parameter is not stored.

Throws:

java.lang.SecurityException - Thrown if the application calling this method does not have an ApplicationStoragePermission with action "manageService" for the organisation_id of this service.

Since:

MHP1.1.2.

remove(AppID)

```
public void remove(org.dvb.application.AppID appId)
throws UserRejectedInstallException
```

Removes a stored application from this service.

Applications should be prepared for the platform consulting the end user of the GEM terminal for the permission to remove the application

If the application identified by the AppID passed in as a parameter is not installed in this service, and the caller has the permissions that would be needed to remove the application if it was installed, the method shall fail silently.

Successfully removing an application from a stored service that is currently being presented shall cause the terminal to send an AppsDatabaseEvent.APP_DELETED event to registered listeners, and if the application is Loaded, Active or Paused then it shall be destroyed as if it had been signalled as KILL in the AIT. (Note that there is no special-case for if an application removes itself from a stored service - this is not an error and shall be handled normally).

Parameters:

appId - AppID of the application to be removed.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 686 of 861

Throws:

UserRejectedInstallException - If the user chose not to remove the application.

java.lang.SecurityException - Thrown if the application calling this method does not have an ApplicationStoragePermission with action "removeService" for the organisation_id of the application to be removed, and an ApplicationStoragePermission with action "manageService" for the organisation_id of this service.

Since:

MHP1.1.2.

remove(AppID[])

```
public void remove(org.dvb.application.AppID[] appIds)
throws UserRejectedInstallException
```

Removes multiple stored applications from this service.

Applications should be prepared for the platform consulting the end user of the GEM terminal for permission to remove the applications

If an application identified by the AppIDs passed in as a parameter is not installed in this service, and the caller has the permissions that would be needed to remove the application if it was installed, then this method shall ignore that AppID.

This method either succeeds or fails completely. It will never remove some installed applications but not other installed applications.

Successfully removing application(s) from a stored service that is currently being presented shall cause the terminal to send an AppsDatabaseEvent.APP_DELETED event for each application to registered listeners, and if any of the removed applications are Loaded, Active or Paused then they shall be destroyed as if they had been signalled as KILL in the AIT. (Note that there is no special-case for if an application removes itself from a stored service - this is not an error and shall be handled normally).

Parameters:

appIds - AppIDs of the applications to be removed.

Throws:

java.lang.IllegalArgumentException - If the array passed to this method has length zero.

java.lang.SecurityException - thrown if the application calling this method does not have ApplicationStoragePermission s with action "removeService" for the organisation_ids of all the applications to be removed, and an ApplicationStoragePermission with action "manageService" for the organisation_id of this service.

UserRejectedInstallException - If the user chose not to remove the application.

Since:

MHP1.1.2.

store(AppProxy[], boolean[], String[][])

```
public void store(org.dvb.application.AppProxy[] apps, boolean[] autoStart,
java.lang.String[][] args)
throws InvalidApplicationException, UserRejectedInstallException, NotEnoughResourcesException, Inval
idDescriptionFileException, ApplicationDownloadException
```

Installs several applications into this stored service.

If ApplicationStorageHandler is supported by the implementation and such a handler is registered then the handler will be asked for permission to install the application. If no such handler is asked then the end user of the GEM terminal may be asked for permission. Applications should be prepared for the possibility of a user interface being shown under these circumstances.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 687 of 861

NOTE: This method is synchronous and will block until the installation is either completed or fails.

This method either succeeds or fails completely. It will never install some applications but not others.

For each specified application, if that application is already installed in the service (whether the same version or a different version, then the rules specified in the `store()` method that stores a single application shall apply regarding updating the application and/or the stored representation of the AIT for that application. Note that the preceeding statement that this method either completely succeeds or completely fails still applies.

If this service is being presented, then on success the rules in the `store()` method that stores a single application regarding sending `AppsDatabaseEvent`s and starting and killing applications shall apply.

Parameters:

`apps` - an array of AppProxys representing the applications to be installed. May not be `null` or zero-length.

`autoStart` - An array that is the same length as `apps` where each element is `true`, if the corresponding application in `apps` becomes an autostart application in the stored application service; or `false`, if the application becomes a normal present (non-autostart) application. May not be `null`.

`args` - parameters to be available to the applications when started. May be `null`, indicating all applications take no parameters. Otherwise it is an array that is the same length as `apps`, where each element is the arguments for the corresponding application in `apps`. If the element is either null or a subarray of size zero, that indicates no parameters are to be available. These parameters shall be available to applications when running as part of the stored application service using the Xlet property "dvb.installer.parameters".

Throws:

`java.lang.IllegalArgumentException` - If the arrays passed to this method have different lengths, or if they have length zero.

`InvalidApplicationException` - thrown if any of the applications do not include an application_storage_descriptor, or if any of the applications are not identified as able to run stand-alone in their application_storage_desciptor.

`InvalidDescriptionFileException` - thrown if one of the application description files is missing, invalid or otherwise not conformant to the specification.

`NotEnoughResourcesException` - thrown if the GEM terminal does not have enough resources, e.g. storage space, available for the applications.

`UserRejectedInstallException` - thrown if either the end user or an ApplicationStorageHandler rejects the installation.

`ApplicationDownloadException` - thrown if the downloading of the application files was notsuccessful (e.g. a carousel error, a file in the application description file is missing in the carousel, if the application was removed from the AIT or from its transport mechanism while installation is in progress, etc) or if the application failed authentication while being downloaded.

`java.lang.SecurityException` - thrown if the application calling this method does not have `ApplicationStoragePermission` s with action "storeService" for the organisation_ids of all the applications to be stored, and an `ApplicationStoragePermission with action "manageService" for the organisation_id of this service.`

Since:

MHP1.1.2.

store(AppProxy[], boolean[], String[][], ApplicationStorageListener)

```
public void store(org.dvb.application.AppProxy[] apps, boolean[] autoStart,
java.lang.String[][] args, org.dvb.application.storage.ApplicationStorageListener listener)
```

Installs several applications into this stored service.

If ApplicationStorageHandler is supported by the implementation and such a handler is registered then the handler will be asked for permission to install the application. If no such handler is asked then the end user of the GEM terminal may be asked for permission. Applications should be prepared for the possibility of a user interface being shown under these circumstances.

This method is asynchronous with completion reported via the ApplicationStorageListener.

This method either succeeds or fails completely. It will never install some applications but not others.

For each specified application, if that application is already installed in the service (whether the same version or a different version, then the rules specified in the `store()` method that stores a single application shall apply regarding updating the application and/or the stored representation of the AIT for that application. Note that the preceeding statement that this method either completely succeeds or completely fails still applies.

If this service is being presented, then on success the rules in the `store()` method that stores a single application regarding sending `AppsDatabaseEvent`s and starting and killing applications shall apply.

Parameters:

`apps` - an array of AppProxys representing the applications to be installed. May not be `null` or zero-length.

`autoStart` - An array that is the same length as `apps` where each element is `true`, if the corresponding application in `apps` becomes an autostart application in the stored application service; or `false`, if the application becomes a normal present (non-autostart) application. May not be `null`.

`args` - parameters to be available to the applications when started. May be `null`, indicating all applications take no parameters. Otherwise it is an array that is the same length as `apps`, where each element is the arguments for the corresponding application in `apps`. If the element is either null or a subarray of size zero, that indicates no parameters are to be available. These parameters shall be available to applications when running as part of the stored application service using the Xlet property "dvb.installer.parameters".

`listener` - the listener to be notified of the success or failure of this installation.

Throws:

`java.lang.IllegalArgumentException` - If the arrays passed to this method have different lengths, or if they have length zero.

`java.lang.SecurityException` - thrown if the application calling this method does not have `ApplicationStoragePermission` s with action "storeService" for the organisation_ids of all the applications to be stored, and an `ApplicationStoragePermission with action "manageService" for the organisation_id of this service.`

Since:

MHP1.1.3.

store(AppProxy, boolean, String[])

```
public void store(org.dvb.application.AppProxy app, boolean autoStart, java.lang.String[] args)
throws InvalidApplicationException, UserRejectedInstallException, NotEnoughResourcesException, Inval
idDescriptionFileException, ApplicationDownloadException
```

Installs an application into this stored service.

If ApplicationStorageHandler is supported by the implementation and such a handler is registered then the handler will be asked for permission to install the application. If no such handler is asked then the end user of the GEM terminal may be asked for permission. Applications should be prepared for the possibility of a user interface being shown under these circumstances.

NOTE: This method is synchronous and will block until the installation is either completed or fails.

Successfully adding an application to a stored service that is currently being presented shall cause the terminal to behave as if the application was added to the AIT - i.e. it will send an `org.dvb.application.AppsDatabaseEvent.APP_ADDED` event to registered listeners, and if the `autoStart` parameter is `true` then the application shall be started.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 689 of 861

If the same version of the same application is already installed in this stored service, then (if the caller has sufficient permissions to install the application) this method shall update the stored representation of the AIT from the `autoStart` and `args` parameters, and return without throwing an exception. Terminals shall not store the application again. Terminals are not expected (but are allowed) to prompt the user for permission in this case. If the stored service is currently being presented, and the autostart setting is changed, then:

- An `org.dvb.application.AppsDatabaseEvent.APP_CHANGED` event shall be sent as usual to registered listeners.
- If the autostart setting is changed to `true`, then the application shall be started as usual. (Note that the opposite is not true: setting autostart to `false` shall not cause the application to be killed if it is running).

Note that calling this method with a new value for the `args` parameter shall not have any effect on an existing instance of an application.

If a different version of the same application is already installed in this stored service, then this method shall install the new version of the application as normal. (Note that for the purposes of this method description, "new" and "old" versions refer to the version specified by the AppProxy passed to this method and the version currently installed in the service, respectively. They do not refer to numeric comparison of version numbers). On success, the old version shall be removed from the service. On failure, the old version of the application shall remain as it was before this method was called. If the stored service is currently being presented, then on success:

- An `org.dvb.application.AppsDatabaseEvent.APP_CHANGED` event shall be sent as usual to registered listeners.
- If the old version of the application is running (or paused) then it shall be terminated as if it had been signalled as KILL in the AIT.
- If the new version is signalled as autostart, then the new version shall be started as usual, once the old version has terminated.

Parameters:

`app` - an AppProxy representing the application to be installed.

`autoStart` - true, if the application becomes an autostart application in the stored application service; false, if the application becomes a normal present (non-autostart) application in the stored application service.

`args` - parameters to be available to the application when started. Passing in either null or an array of size zero indicates no parameters are to be available. These parameters shall be available to applications when running as part of the stored application service using the Xlet property "dvb.installer.parameters".

Throws:

`InvalidApplicationException` - thrown if the application does not include an application_storage_descriptor, or if the application is not identified as able to run stand-alone in it's application_storage_desciptor.

`InvalidDescriptionFileException` - thrown if the application description file is missing, invalid or otherwise not conformant to the specification.

`NotEnoughResourcesException` - thrown if the GEM terminal does not have enough resources, e.g. storage space, available for the application.

`UserRejectedInstallException` - thrown if either the end user or an ApplicationStorageHandler rejects the installation.

`ApplicationDownloadException` - thrown if the downloading of the application files was notsuccessful (e.g. a carousel error, a file in the application description file is missing in the carousel, if the application was removed from the AIT or from its transport mechanism while installation is in progress, etc) or if the application failed authentication while being downloaded.

`java.lang.SecurityException` - thrown if the application calling this method does not have an `ApplicationStoragePermission` with action "storeService" for the organisation_id of the application to be stored, and an `ApplicationStoragePermission` with action "manageService" for the organisation_id of this service.

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

Since:

MHP1.1.2.

store(Locator, AppID[], boolean[], String[][], ApplicationStorageListener)

```
public void store(org.davic.net.Locator locator, org.dvb.application.AppID[] apps,
boolean[] autoStart, java.lang.String[][] args,
org.dvb.application.storage.ApplicationStorageListener listener)
```

Installs several applications into this stored service. The applications are from a service which need not be selected.

If ApplicationStorageHandler is supported by the implementation and such a handler is registered then the handler will be asked for permission to install the application. If no such handler is asked then the end user of the GEM terminal may be asked for permission. Applications should be prepared for the possibility of a user interface being shown under these circumstances.

This method is asynchronous with completion reported via the ApplicationStorageListener.

This method either succeeds or fails completely. It will never install some applications but not others.

For each specified application, if that application is already installed in the service (whether the same version or a different version, then the rules specified in the `store()` method that stores a single application shall apply regarding updating the application and/or the stored representation of the AIT for that application. Note that the preceeding statement that this method either completely succeeds or completely fails still applies.

If this service is being presented, then on success the rules in the `store()` method that stores a single application regarding sending `AppsDatabaseEvent`s and starting and killing applications shall apply.

Parameters:

`locator` - the locator of the service from which the applications are to be stored.

`apps` - an array of application ids identifying the applications to be stored.

`autoStart` - An array that is the same length as `apps` where each element is `true`, if the corresponding application in `apps` becomes an autostart application in the stored application service; or `false`, if the application becomes a normal present (non-autostart) application. May not be `null`.

`args` - parameters to be available to the applications when started. May be `null`, indicating all applications take no parameters. Otherwise it is an array that is the same length as `apps`, where each element is the arguments for the corresponding application in `apps`. If the element is either null or a subarray of size zero, that indicates no parameters are to be available. These parameters shall be available to applications when running as part of the stored application service using the Xlet property "dvb.installer.parameters".

`listener` - the listener to be notified of the success or failure of this installation.

Throws:

`java.lang.IllegalArgumentException` - If the arrays passed to this method have different lengths, or if they have length zero.

`java.lang.SecurityException` - thrown if the application calling this method does not have `ApplicationStoragePermission` s with action "storeService" for the organisation_ids of all the applications to be stored, and an `ApplicationStoragePermission with action "manageService" for the organisation_id of this service.`

Since:

MHP1.1.3.

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

org.dvb.application.storage

ApplicationStorageListener

Declaration

```
public interface ApplicationStorageListener
```

Description

Listener to receive reports on progress of application storage operations. When multiple applications are being stored, calls to this listener shall be generated for each individual application.

Methods

storageCompleted(AppID)

```
public void storageCompleted(org.dvb.application.AppID id)
```

Called when a storage operation is successfully completed.

Parameters:

`id` - the application which is now successfully stored.

storageFailed(AppID, Exception)

```
public void storageFailed(org.dvb.application.AppID id, java.lang.Exception failureMode)
```

Called if a storage operation fails.

Parameters:

`id` - the application whose storage failed.

`failureMode` - the reason for the failure as would be reported by one of the checked exceptions thrown by the method StoredApplicationService.store.

storageUpdate(AppID, byte, int)

```
public void storageUpdate(org.dvb.application.AppID id, byte completion, int size)
```

Called at 1 second intervals after a storage operation starts until it completes. The precise definition of completeness is implementation dependent however it would typically reflect the extent to which the data of the application has been downloaded. In implementations where an application is first downloaded and then stored, assuming the storage phase takes much less time than the downloading phase, the storage phase can be ignored when reporting completeness. If an application is (already) downloaded and stored before the first call to this method becomes due, this method may never be called.

Parameters:

`id` - the application being stored.

`size` - the size in bytes of the application to be stored, as calculated by adding the size attributes in the application description file of the application.

`completion` - the extent to which the storage operation is complete, 0 being not complete and 255 being almost complete.

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

org.dvb.application.storage

ApplicationStoragePermission

Declaration

```
public class ApplicationStoragePermission extends java.security.Permission

java.lang.Object
  |
  +--java.security.Permission
        |
        +--org.dvb.application.storage.ApplicationStoragePermission
```

All Implemented Interfaces:

```
java.security.Guard, java.io.Serializable
```

Description

This class represents a permission to manage applications stored in the GEM terminal. An ApplicationStorage-Permission contains a name representing the organisation_id whose applications can be managed and an actions list representing the permitted actions, e.g. store and/or remove applications.

The name of the permission contains the organisation_id represented in hexadecimal form as defined in the section "Text encoding of application identifiers" in the main body of this specification. Valid organization ids must be in the range "1" to "ffffffff" inclusive. Alternatively, the value "*" indicates all organization ids.

The actions string shall be a comma-separated list of one or more of the following:

- "manageService", representing permission to query what applications are stored in a stored application service with the given organisation ID. This permission is also necessary (but not sufficient) to store applications into and remove applications from a stored application service, where the stored application service has the given organisation ID.
- "storeService", representing permission to store an application in a stored application service, where the application has the given organisation ID.
- "removeService", representing permission to remove an application from a stored application service, where the application has the given organisation ID.
- "createService", representing permission to create a stored application service with a given organisation ID.
- "deleteService", representing permission to remove a stored application service with a given organisation ID.
- "manageCache", representing permission to query what applications are stored in an application cache with the given organisation ID. This permission is also necessary (but not sufficient) to store applications into and remove applications from an application cache, where the application cache has the given organisation ID.
- "storeCache", representing permission to store an application in a cache.
- "removeCache", representing permission to remove an application from a cache.

Since:

MHP1.1.

Constructors

ApplicationStoragePermission(String, String)

```
public ApplicationStoragePermission(java.lang.String name, java.lang.String actions)
```


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

Creates a new ApplicationStoragePermission object with the specified name and actions string. The name contains the organisation_id of the applications that can be managed and the actions String shall be a comma-separated list of actions as defined above. Permission objects constructed with incorrectly encoded parameters do not represent any permission and are ignored by the platform.

Parameters:

name - the organisation_id whose applications can be managed. This is encoded in hexadecimal representation as if by java.lang.Integer.toHexString(int) , and must be in the range "1" to "ffffffff" inclusive. Alternatively, the value "*" indicates all organization ids.

actions - Shall conform to the syntax described above.

Methods

equals(Object)

```
public boolean equals(java.lang.Object obj)
```

Checks two permission objects for equality.

Do not use the equals method for making access control decisions; use the implies method.

If X is an ApplicationStoragePermission, and Y is any Object, then X.equals(Y) returns true if and only if all of the following hold:

- Y is not null.
- Y is an instance of ApplicationStoragePermission.
- X and Y have the same run-time type (i.e. X.getClass() == Y.getClass()).
- X and Y were both constructed with correctly encoded parameters.
- X and Y both have the same organization ID, or both have organization ID "*".
- X has the same actions as Y. The order of the comma-separated actions list does not affect the results of this check.

Overrides:

equals in class Permission

Parameters:

obj - the object we are testing for equality with this object.

Returns:

true if both ApplicationStoragePermission objects are equivalent.

getActions()

```
public java.lang.String getActions()
```

Returns the actions as a String. Must always return actions in canonical form.

Permission objects constructed with an incorrectly encoded action parameter shall return an empty string.

Permission objects constructed with a correctly encoded action parameter shall return a comma-separated list of actions, with the actions sorted in the order given by java.lang.String.compareTo(String) .

Overrides:

getActions in class Permission

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 694 of 861

Returns:

the actions of this `Permission`.

hashCode()

```
public int hashCode()
```

Returns the hash code value for this `ApplicationStoragePermission` object.

The required `hashCode` behavior for `ApplicationStoragePermission` objects is the following:

- Whenever it is invoked on the same ApplicationStoragePermission object more than once during an execution of a Java application, the `hashCode` method must consistently return the same integer. This integer need not remain consistent from one execution of an application to another execution of the same application.
- If two `ApplicationStoragePermission` objects are equal according to the `equals(Object)` method, then calling the `hashCode` method on each of the two Permission objects must produce the same integer result.

Overrides:

`hashCode` in class `Permission`

Returns:

a hash code value for this object.

implies(Permission)

```
public boolean implies(java.security.Permission permission)
```

Checks if the specified permission's actions are "implied by" this object's actions.

If `X` is an `ApplicationStoragePermission`, and `Y` is any `Permission`, then `X.implies(Y)` returns true if and only if all of the following hold:

- Y is an instance of ApplicationStoragePermission.
- X and Y have the same run-time type (i.e. X.getClass() == Y.getClass()).
- X and Y were both constructed with correctly encoded parameters.
- X and Y both have the same organization ID, or X has the organization ID "*".
- X contains all the actions requested by Y. The order of the comma-separated actions list does not affect the results of this check.

Overrides:

`implies` in class `Permission`

Parameters:

`permission` - the permission to check against.

Returns:

`true` if the specified permission is implied by this object, `false` if not.

newPermissionCollection()

```
public java.security.PermissionCollection newPermissionCollection()
```

Returns an empty `PermissionCollection` for `ApplicationStoragePermission` objects.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 695 of 861

Overrides:

`newPermissionCollection` in class `Permission`

Returns:

a new `PermissionCollection` object for `ApplicationStoragePermissions`.

org.dvb.application.storage ExtendedAppAttributes

Declaration

```
public interface ExtendedAppAttributes extends org.dvb.application.AppAttributes
```

All Superinterfaces:

`org.dvb.application.AppAttributes`

Description

The `ExtendedAppAttributes` interface provides additional attributes that are useful when application can be stored in the GEM terminal.

`org.dvb.application.AppAttributes` objects that are returned from the `org.dvb.application.AppsDatabase` shall implement this interface if and only if the terminal supports storing applications and the total amount of memory for stored services and/or cached applications is greater than zero.

Since:

MHP1.1.

Methods

canAddToStoredService()

```
public boolean canAddToStoredService()
```

Returns `true` if this application is signalled as capable of being added to a stored service. I.e. if this application could be added to a stored service without the `InvalidApplicationException` being thrown.

For broadcast applications, returns `true` if and only if this application is signalled as being storable and as capable of running stand alone. For applications that are part of a stored service, this function returns `true`.

Returns:

`true` if this application is signalled as capable of being added to a stored service.

Since:

MHP1.1.2.

canCache()

```
public boolean canCache()
```

Returns `true` if this application is signalled as capable of being cached. I.e. if this application could be added to a cache without the `InvalidApplicationException` being thrown.

For broadcast applications, returns `true` if and only if this application is signalled as being storable. For applications that are part of a stored service, this function returns `true`.

Returns:

`true` if this application is signalled as capable of being cached.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 696 of 861

Since:

MHP1.1.2.

getCurrentVersionNumber()

```
public int getCurrentVersionNumber()
```

Returns the optional version number currently signalled for this application. (E.g. in the AIT). If no version number is signalled, -1 shall be returned.

Returns:

the version number signalled for this application.

Since:

MHP1.1.

isStartable()

```
public boolean isStartable()
```

This method determines whether the application is startable or not. An Application is not startable if any of the following apply.

- The application is transmitted on a remote connection, and either does not have an application storage descriptor, is not cached, or is not signalled as launchable completely from cache.
- The caller of the method does not have the Permissions to start it.
- if the application is signalled with a control code which is neither AUTOSTART nor PRESENT.
- if the application is signalled with an application storage descriptor where not_launchable_from_broadcast is "1", and the application is not stored or cached.

If none of the above apply, then the application is startable.

The value returned by this method does not depend on whether the application is actually running or not.

Overrides:

`isStartable` in interface `AppAttributes`

Returns:

true if an application is startable, false otherwise.

Since:

MHP1.0.

isStorageRequired()

```
public boolean isStorageRequired()
```

Returns `true` if this application is signalled as not launchable from broadcast. I.e. for a broadcast application this method will return `true` if and only if the application is signalled with an application storage descriptor where not_launchable_from_broadcast is "1". If this application is part of a stored service, this function returns `true`.

Note that the return value of this method does not depend on whether or not the application is currently stored or cached.

Returns:

`true` if and only if this application is signalled as not launchable from broadcast.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 697 of 861

Since:

MHP1.1.2.

isStored()

```
public boolean isStored()
```

Returns `true` if this application is currently stored or cached on the terminal. Returns `true` if and only if the application is signalled as being storable and all the files signalled as having critical priority have been stored or cached by the terminal.

Returns:

`true` if this application is currently stored or cached.

Since:

MHP1.1.2.

org.dvb.application.storage InvalidApplicationException

Declaration

```
public class InvalidApplicationException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.application.storage.InvalidApplicationException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

When an application is being installed into a service, this is thrown when:

- The application does not include an application_storage_descriptor.
- The application is not identified as able to run stand-alone in its application_storage_descriptor.

When an application is being stored in a cache this is thrown when:

- The application does not include an application_storage_descriptor.
- The application is not signalled as part of the same service as the calling application.

Since:

MHP1.1.

Constructors

InvalidApplicationException()

```
public InvalidApplicationException()
```

Constructs an `InvalidApplicationException` with no detail message.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 698 of 861

InvalidApplicationException(String)

```
public InvalidApplicationException(java.lang.String s)
```

Construct a `InvalidApplicationException` with a detail message.

Parameters:

`s` - detail message.

org.dvb.application.storage InvalidDescriptionFileException

Declaration

```
public class InvalidDescriptionFileException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.application.storage.InvalidDescriptionFileException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown when an application descriptio file is invalid.

Constructors

InvalidDescriptionFileException()

```
public InvalidDescriptionFileException()
```

Creates a new instance of `InvalidDescriptionFileException` without detail message.

InvalidDescriptionFileException(String)

```
public InvalidDescriptionFileException(java.lang.String msg)
```

Constructs an instance of `InvalidDescriptionFileException` with the specified detail message.

Parameters:

`msg` - the detail message.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 699 of 861

org.dvb.application.storage NotEnoughResourcesException

Declaration

```
public class NotEnoughResourcesException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.application.storage.NotEnoughResourcesException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown if the GEM terminal does not have enough resources to install the application. E.g. not enough storage space.

Since:

MHP1.1.

Constructors

NotEnoughResourcesException()

```
public NotEnoughResourcesException()
```

Constructs a `NotEnoughResourcesException` with no detail message.

NotEnoughResourcesException(String)

```
public NotEnoughResourcesException(java.lang.String s)
```

Constructs a `NotEnoughResourcesException` with a detail message

Parameters:

`s` - detail message.

org.dvb.application.storage ServiceAlreadyExistsException

Declaration

```
public class ServiceAlreadyExistsException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.application.storage.ServiceAlreadyExistsException
```

All Implemented Interfaces:

```
java.io.Serializable
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 700 of 861

Description

Thrown if a StoredApplicationService already exists with the identifiers the application tried to create a new one.

Since:

MHP1.1.

Constructors

ServiceAlreadyExistsException()

```
public ServiceAlreadyExistsException()
```

Constructs a `ServiceAlreadyExistsException` with no detail message.

ServiceAlreadyExistsException(String)

```
public ServiceAlreadyExistsException(java.lang.String s)
```

Constructs a `ServiceAlreadyExistsException` with a detail message.

Parameters:

`s` - detail message.

org.dvb.application.storage

StoredApplicationService

Declaration

```
public interface StoredApplicationService extends javax.tv.service.Service,
ApplicationStorageController
```

All Superinterfaces:

`ApplicationStorageController`, `javax.tv.service.Service`

Description

Defines the information about a stored application service.

Stored application services can be created by applications that have a permission to store applications for an organisation_id.

Stored application services are uniquely identified within the terminal by the combination of the organisation_id and service_id.

Since:

MHP1.1.

Methods

getLocator()

```
public javax.tv.locator.Locator getLocator()
```

Gets the locator for this stored application service. This locator is opaque and platform specific. It shall be an instance of `org.davic.net.Locator` or a subclass. It is not required to be an instance of any other public locator class in the platform (e.g. `org.davic.net.dvb.DvbLocator`).

Overrides:

`getLocator` in interface `Service`

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

Returns:

a locator for this stored application service.

getName()

```
public java.lang.String getName()
```

Returns the name of the stored application service as defined when the stored application service was created.

Overrides:

`getName` in interface `Service`

Returns:

the name of the stored application service.

getOrganisationId()

```
public int getOrganisationId()
```

Return the organisation id of this stored application service.

Returns:

the organisation id of this stored application service.

getServiceId()

```
public int getServiceId()
```

Return the service id of this stored application service.

Returns:

the service id of this stored application service.

getServiceInformationType()

```
public javax.tv.service.ServiceInformationType getServiceInformationType()
```

Returns the service information format of this object. This shall always return `ServiceInformationType.UNKNOWN`.

Returns:

the service information format.

getServiceType()

```
public javax.tv.service.ServiceType getServiceType()
```

Returns the type of this service. For stored application services, this method shall always return `StoredApplicationServiceType.STORED_APPLICATION_SERVICE` .

Overrides:

`getServiceType` in interface `Service`

Returns:

service type of this service.

hasMultipleInstances()

```
public boolean hasMultipleInstances()
```

Indicates whether the service represented by this Service is available on multiple transports. For stored application services, this shall always return false.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 702 of 861

Overrides:

hasMultipleInstances in interface Service

Returns:

false.

isSelectable()

```
public boolean isSelectable()
```

Test whether this service is selectable. StoredApplicationServices are selectable if they contain at least one autostart application and are not selectable if they do not contain any autostart applications. Applications which offer the end-user a choice of services to select should not include services which are not selectable in that list.

Returns:

false if the service is not selectable otherwise true.

Since:

MHP 1.1.3.

removeService()

```
public void removeService()
throws UserRejectedInstallException
```

Remove the whole stored application service from the terminal. Removal of the service results in the removal of all the applications from within that service.

If there are any applications installed in the stored application service, then the application calling this method should be prepared for the platform consulting the end user of the GEM terminal for permission to remove the stored service.

The platform shall not prompt the user for permission if no applications are installed in the service.

If the end user is asked and does not give permission to remove the service, none of the applications in the service shall be removed.

Throws:

java.lang.SecurityException - if the caller does not have an ApplicationStoragePermission with action "deleteService" and an organisation_id which matches that of this service. Also thrown if the caller does not have ApplicationStoragePermission with action "removeService" for the organisation IDs of every application installed in this service.

UserRejectedInstallException - If the user chose not to remove the applications in the service.

retrieveDetails(SIRequestor)

```
public javax.tv.service.SIRequest retrieveDetails(javax.tv.service.SIRequestor requestor)
```

This method retrieves additional information about the service. This shall result in failure with the SIRequestFailure-Type DATA_UNAVAILABLE.

Overrides:

retrieveDetails in interface Service

Parameters:

requestor - The SIRequestor to be notified when this operation completes.

Returns:

A SIRequest object identifying this request.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 703 of 861

org.dvb.application.storage StoredApplicationServiceFactory

Declaration

```
public abstract class StoredApplicationServiceFactory

java.lang.Object
  |
  +--org.dvb.application.storage.StoredApplicationServiceFactory
```

Description

This factory creates new Service objects representing stand-alone stored application services. Services thus created shall appear in the list of services maintained by the SIManager until removed using `StoredApplicationService.remove` or some GEM terminal specific mechanism. i.e. they 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.

Since:

MHP1.1.2.

Constructors

StoredApplicationServiceFactory()

```
protected StoredApplicationServiceFactory()
```

This constructor is provided for the use of implementations and specifications which extend GEM. 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

createStoredApplicationService(int, int, String)

```
public abstract org.dvb.application.storage.StoredApplicationService
createStoredApplicationService(int organisation_id, int service_id, java.lang.String serviceName)
throws ServiceAlreadyExistsException
```

Creates a new stored application service.

Parameters:

`organisation_id` - the organisation_id of the organisation to whom this service belongs to

`service_id` - unique identifier for this service within the organisation

`serviceName` - a name for the service that can be displayed to the end user to identify this service

Returns:

the stored application service created.

Throws:

`serviceAlreadyExistsException` - thrown if a stored application service with the same organisation_id and service_id already exists in the terminal.

`java.lang.SecurityException` - thrown if the application calling this method does not have an Application-StoragePermission with action "createService" for the organisation_id passed in as the parameter.

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

getInstance()

```
public static org.dvb.application.storage.StoredApplicationServiceFactory getInstance()
```

Get the singleton instance of this class, or null if and only if this GEM implementation does not support stand-alone stored applications.

Returns:

a factory.

org.dvb.application.storage StoredApplicationServiceType

Declaration

```
public class StoredApplicationServiceType extends javax.tv.service.ServiceType
```

```
java.lang.Object
  |
  +--javax.tv.service.ServiceType
        |
        +--org.dvb.application.storage.StoredApplicationServiceType
```

Description

The service type used for stored application services.

Since:

MHP 1.1.2.

See Also:

`javax.tv.service.navigation.ServiceTypeFilter`

Fields

STORED_APPLICATION_SERVICE

```
public static final javax.tv.service.ServiceType STORED_APPLICATION_SERVICE
```

The service type for a stored application.

Since:

MHP 1.1.2.

Constructors

StoredApplicationServiceType(String)

```
protected StoredApplicationServiceType(java.lang.String name)
```

Creates a new StoredServiceType object.

Parameters:

`name` - The string name of this type (e.g. "STORED_APPLICATION_SERVICE").

Since:

MHP 1.1.2.

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

org.dvb.application.storage
UserRejectedInstallException

Declaration

```
public class UserRejectedInstallException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.application.storage.UserRejectedInstallException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown if the request to install an application was rejected. This includes both rejection by the end user and rejection by an ApplicationStorageHandler.

Since:

MHP1.1.

Constructors

UserRejectedInstallException()

```
public UserRejectedInstallException()
```

Constructs a `UserRejectedInstallException` with no detail message.

UserRejectedInstallException(String)

```
public UserRejectedInstallException(java.lang.String s)
```

Constructs a `UserRejectedInstallException` with a detail message

Parameters:

`s` - detail message.

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

Annex AH (normative): Internet client APIs

Package org.dvb.internet

Description

Provides a mechanism for GEM applications to control internet clients that may be present on a GEM such as a web browser or email client.

This package provides a mechanism for GEM applications to control internet clients that may be present on a GEM such as a web browser or email client. The API makes no assumptions about whether an internet client and downloaded GEM application can run simultaneously, and in some cases starting an internet client may result in the calling application being killed.

The API consists of two main class hierarchies. `InternetClientService` and its clients provide a mechanism to get a locator for a resident internet client so that it can be started, and also provide any functionality that does not require the client to be started. This can include adding bookmarks or getting the capabilities of the clients. The second major part of the API is the `InternetClient` class and its subclasses. These provide an interface to a running instance of an internet client and support the operations that can only be carried out when the client is running.

The internet client API uses the JavaTV service navigation API for accessing resident applications. Any effect that the invocation of a resident internet client has on the lifecycle of a downloaded application is dependent on the service context in which the resident application is started (using `ServiceContext.select()`). If the internet client is started in the same service context as the downloaded application, then the downloaded application will be terminated, following the usual semantics of the service selection API. The basic concepts are introduced below and then a few scenarios are explored to illustrate combinations or broadcast application versus resident application execution.

Concepts

The first implication of resident applications is that it should be feasible for broadcast applications to isolate, browse, and select such services. The collection of `javax.tv.service.*` packages provides the framework for the solution to this requirement.

The class `ServiceType` in the `javax.tv.service` package defines various kinds of services. This is extended in this API to provide additional service types for Internet clients. The `javax.tv` framework lets a client filter the services known to a platform to create a `ServiceCollection`. The client would first create an instance of `ServiceType-Filter`.

```
public ServiceTypeFilter(ServiceType type)
```

where the `ServiceType` in the signature would be the instance for the resident service in question. The client would then filter the services known to the platform to isolate the resident services of this type. The `ServiceCollection` interface provides the filter operation.

```
public ServiceCollection createServiceCollection(ServiceFilter filter)
```

Given the `ServiceCollection`, the client can iterate over the resident services to find a specific resident service. The last step is to `select()` the resident service. The `ServiceContext` interface of the `javax.tv.service.selection` package provides the operation:

```
public void select(Service selection)
throws java.lang.SecurityException
```

The select() method is asynchronous and returns before attempting to allocate resources to execute the service.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 707 of 861

If the implementation determines that it can not realize the service due to scarce resources, it forwards a `SelectionFailedEvent` (of the `javax.tv.service. selection` package) with the reason code `INSUFFICIENT_RESOURCES`. The `ServiceContextListener` interface fields the event:

```
public void receiveServiceContextEvent(ServiceContextEvent e)
```

Thus the client should implement an instance of the `ServiceContextListener` interface and register interest in service context events. The `ServiceContext` interface provides the mechanism:

```
public void addListener(ServiceContextListener listener)
throws java.lang.IllegalStateException
```

The semantics of a `ServiceContext` are that at most a single `Service` can execute within a `ServiceContext` at a time. Since the premise is that a broadcast service selects a resident service, there are two scenarios of interest.

The first scenario presumes that the broadcast service wants to select the resident service but not survive. It expects the selection to cause its termination. The broadcast service in this case should create an instance of `ServiceContextFactory`. The operation on `ServiceContextFactory` itself is:

```
public static ServiceContextFactory getInstance()
```

The broadcast application then requests the service contexts that are visible to the broadcast application:

```
public abstract ServiceContext getServiceContext(XletContext ctx)
```

which returns *its* `ServiceContext`. The broadcast application selects the resident application within its service context:

```
public void select(Locator selection)
throws InvalidLocatorException, java.lang.SecurityException,
java.lang.IllegalStateException
```

which causes the broadcast application to terminate.

The premise of the second scenario is that the broadcast application wants to survive the selection of the resident service. In this case the broadcast again creates a `ServiceContextFactory` but then creates a `ServiceContext` which is not its `ServiceContext`:

```
public abstract ServiceContext createServiceContext()
throws InsufficientResourcesException,
java.lang.SecurityException
```

The broadcast application selects the resident application. Since the operation is on a `ServiceContext` that is not its `ServiceContext`, the broadcast application survives the selection.

One subtle aspect of the invocation patterns is that the platform does not know whether the broadcast application will attempt to select a resident application versus another broadcast application until the broadcast application invokes the `select()` operation. For example, assume the platform supports a single broadcast application plus a single resident application. Further assume that, after the broadcast application creates a second `ServiceContext`, it selects a second broadcast application rather than a resident application. If the locator is valid, the operation succeeds, but the broadcast application then receives a `SelectionFailedEvent`.

Likewise assume the platform supports multiple broadcast applications but not (for the present) a resident application. If the broadcast application selects a resident application, and if the locator is valid, the operation succeeds, but the broadcast application then receives a `SelectionFailedEvent`. (The second case is rather improbable. If the platform does not support resident applications, it is not clear how the broadcast application could obtain a valid locator to a broadcast application. The platform would have to support resident applications in concept, but not at the instant the broadcast applications selects a resident application.)

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 708 of 861

Class Summary	
Interfaces	
EmailClient	Interface supporting the operations required on an email client.
EmailClientService	Service representing the resident email client
InternetClient	Base interface for the internet clients.
InternetClientListener	Interface for objects that wish to receive InternetClientEvents
InternetClientService	The base class for the interface to resident applications that are supported by the internet access profile.
UsenetClient	This interface supports the operations required on a Usenet news client.
UsenetClientService	Service representing the resident usenet news client
WWWBrowser	This interface provides support for the operations required on an WWW browser.
WWWBrowserService	Service representing a resident WWW browser
Classes	
CancelledByUserEvent	Event indicating that an operation on an internet client failed because the operation was canceled by the user
HomePagePermission	This class is for permissions related to the ability for a GEM application to set the home page of a WWW browser in the internet access profile.
InternetClientEvent	Base class for all status events from internet clients.
InternetClientFailureEvent	Event indicating that an operation on an internet client failed.
InternetClientSuccessEvent	Event indicating that an operation on an internet client succeeded
InternetServiceFilter	InternetServiceFilter represents a service type for a particular kind of internet client.
InternetServiceType	Class representing the additional service types available in the Internet Access profile.
PermissionDeniedEvent	Event indicating that an operation on an internet client failed due to the specified URL not being accessible by the client due to access controls on the server.
URLUnavailableEvent	Event indicating that an operation on an internet client failed due to the specified URL not being available.
Exceptions	
ClientNotRunningException	Exception thrown when a method of InternetClient or one of its subclasses is called while the client is not running, for instance if the client has been terminated by the user.
EntryExistsException	Exception generated when an application tries to add a bookmark or address book entry that already exists

org.dvb.internet

CancelledByUserEvent

Declaration

```
public class CancelledByUserEvent extends InternetClientFailureEvent

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.internet.InternetClientEvent
              |
              +--org.dvb.internet.InternetClientFailureEvent
                    |
                    +--org.dvb.internet.CancelledByUserEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Event indicating that an operation on an internet client failed because the operation was canceled by the user.

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

Constructors

CancelledByUserEvent(Object, URL)

```
public CancelledByUserEvent(java.lang.Object source, java.net.URL url)
```

Construct a new `CancelledByUserEvent`.

Parameters:

`source` - the source of the event.

`url` - the URL to which the event relates.

org.dvb.internet ClientNotRunningException

Declaration

```
public class ClientNotRunningException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.internet.ClientNotRunningException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Exception thrown when a method of `InternetClient` or one of its subclasses is called while the client is not running, for instance if the client has been terminated by the user.

Constructors

ClientNotRunningException()

```
public ClientNotRunningException()
```

Construct a `ClientNotRunningException` with no detail message.

ClientNotRunningException(String)

```
public ClientNotRunningException(java.lang.String reason)
```

Construct a `ClientNotRunningException` with the specified detail message.

Parameters:

`reason` - the reason why the exception was thrown.

org.dvb.internet EmailClient

Declaration

```
public interface EmailClient extends InternetClient
```


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

All Superinterfaces:

InternetClient, javax.tv.service.selection.ServiceContentHandler

Description

Interface supporting the operations required on an email client.

Methods

createMessage(String, String, String)

```
public void createMessage(java.lang.String to, java.lang.String subject,
java.lang.String messageBody)
throws ClientNotRunningException
```

Create a new email message. If any of the parameters are an empty string, the user will be prompted for the missing information.

This is an asynchronous operation, whose success or failure will be indicated by an InternetClientSuccessEvent or InternetClientFailureEvent or one of their subclasses.

Parameters:

to - the address to which the email should be sent.

subject - the subject for the email.

messageBody - the body of the message.

Throws:

java.lang.NullPointerException - if any of the to address, subject or message body are null.

ClientNotRunningException - if the client is not currently running.

java.lang.IllegalArgumentException - if the destination address is an empty string.

org.dvb.internet

EmailClientService

Declaration

```
public interface EmailClientService extends InternetClientService
```

All Superinterfaces:

InternetClientService, javax.tv.service.Service

Description

Service representing the resident email client.

Methods

addToAddressBook(String, String)

```
public void addToAddressBook(java.lang.String address, java.lang.String name)
throws EntryExistsException, IOException
```

Add an entry to the address book. As a side-effect an entry previously added by this method may be lost. Implementations may restrict the number of entries a single GEM application or source of applications may add.

Parameters:

address - the address to be added to the address book.

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

name - the name that should be associated with that address.

Throws:

EntryExistsException - if an entry with both the same name and the same address already exists in the address book.

java.lang.IllegalArgumentException - if the string passed as the email address does not conform to the internet email address format.

IOException - if no more address book entries can be added due to a lack of storage space or a limitation in the client.

java.io.IOException.

getUserEmailAddress()

```
public java.lang.String getUserEmailAddress()
```

Get the email address of the user. The returned value shall be the same as that obtained by reading the value of the "User @" preference (see org.dvb.user.GeneralPreferences for details).

Returns:

the email address of the user, or null if no email address is set or the email address is not available to the client.

Throws:

java.lang.SecurityException - if the caller does not have a UserPreferencePermission with the name "read".

setInitialMessage(String, String, String)

```
public void setInitialMessage(java.lang.String to, java.lang.String subject,
java.lang.String messageBody)
```

Set the initial recipient, subject and& message body to be used when the email client starts. This URL is specific to this instance of EmailClientService and will not impact any other instance and is only valid for the lifetime of this instance. Calling this method and then selecting the EmailClientService instance is equivalent to selecting the EmailClientService instance, obtaining the EmailClient and then calling the createMessage method there. If the application calling this method is still running when the message is sent (or fails) and has registered an InternetClientListener then the appropriate InternetClientEvent shall be sent corresponding to the success or failure of the operation to send the message.

Parameters:

to - the address to which the email should be sent.

subject - the subject for the email.

messageBody - the body of the message.

Throws:

java.lang.NullPointerException - if any of the to address, subject or message body are null.

java.lang.IllegalArgumentException - if the destination address is an empty string.

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

org.dvb.internet EntryExistsException

Declaration

```
public class EntryExistsException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.internet.EntryExistsException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Exception generated when an application tries to add a bookmark or address book entry that already exists.

Constructors

EntryExistsException()

```
public EntryExistsException()
```

Construct a `EntryExistsException` with no detail message.

EntryExistsException(String)

```
public EntryExistsException(java.lang.String message)
```

Construct a `EntryExistsException` with the specified detail message.

Parameters:

`message` - the reason why the exception was thrown.

org.dvb.internet HomePagePermission

Declaration

```
public class HomePagePermission extends java.security.BasicPermission

java.lang.Object
  |
  +--java.security.Permission
        |
        +--java.security.BasicPermission
              |
              +--org.dvb.internet.HomePagePermission
```

All Implemented Interfaces:

```
java.security.Guard, java.io.Serializable
```

Description

This class is for permissions related to the ability for a GEM application to set the home page of a WWW browser in the internet access profile. If an application has this permission then shall be able to set the home page.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 713 of 861

Constructors

HomePagePermission(String)

```
public HomePagePermission(java.lang.String name)
```

Creates a new `HomePagePermission`. The name parameter is not used and must be set to empty string "". Implementations of this version of GEM shall ignore the name, but it could be taken into use in future versions of GEM.

Parameters:

`name` - the name of the `HomePagePermission`. Not used, shall be "".

HomePagePermission(String, String)

```
public HomePagePermission(java.lang.String name, java.lang.String actions)
```

Creates a new `HomePagePermission`. The name and actions parameters are not used. Implementations of this version of GEM shall ignore the name and actions, but they could be taken into use in future versions of GEM. This constructor exists for use by the Policy object to instantiate new Permission objects.

Parameters:

`name` - the name of the `HomePagePermission`. Not used, shall be "".

`actions` - Not used, shall be null.

org.dvb.internet

InternetClient

Declaration

```
public interface InternetClient extends javax.tv.service.selection.ServiceContentHandler
```

All Superinterfaces:

```
javax.tv.service.selection.ServiceContentHandler
```

All Known Subinterfaces:

`EmailClient`, `UsenetClient`, `WWWBrowser`

Description

Base interface for the internet clients. Access to those methods common to all running instances of the client (e.g. operations on bookmark lists) are all carried out through the service objects associated with the `InternetClient` object in question. These are accessed using the `getService()` method.

Methods

addInternetClientListener(InternetClientListener)

```
public void addInternetClientListener(org.dvb.internet.InternetClientListener l)
```

Add a listener for `InternetClientEvents`. If the listener is already registered, or the client is not running, then calls to this method have no effect.

Parameters:

`l` - the listener to be added.

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

getService()

```
public org.dvb.internet.InternetClientService getService()
```

Get the service object which matches this internet client. In the case of a web browser, for example, this would be an instance of the WWWBrowserService class.

Returns:

the service which matches the `InternetClient` object.

getServiceContentLocators()

```
public javax.tv.locator.Locator[] getServiceContentLocators()
```

Reports the portions of the service on which this handler operates.

Overrides:

`getServiceContentLocators` in interface `ServiceContentHandler`

Returns:

An array of length 1, containing the locator representing the internet client. This shall be the same locator returned by calls to `getService().getLocator()`.

removeInternetClientListener(InternetClientListener)

```
public void removeInternetClientListener(org.dvb.internet.InternetClientListener l)
```

Remove a listener for `InternetClientEvents`. If the listener is not registered, or the client is not running, then calls to this method have no effect.

Parameters:

`l` - the listener to be added.

org.dvb.internet

InternetClientEvent

Declaration

```
public class InternetClientEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.internet.InternetClientEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Direct Known Subclasses:

`InternetClientFailureEvent`, `InternetClientSuccessEvent`

Description

Base class for all status events from internet clients.

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

Constructors

InternetClientEvent(Object, URL)

```
public InternetClientEvent(java.lang.Object source, java.net.URL url)
```

Construct a new `InternetClientEvent`.

Parameters:

`source` - the source of the event.

`url` - the URL to which the event relates.

Methods

getUrl()

```
public java.net.URL getUrl()
```

Get the URL for which this event was generated. In the case that a URL was not specified when the event was constructed, null shall be returned.

Returns:

an instance of `java.net.URL` or null.

org.dvb.internet InternetClientFailureEvent

Declaration

```
public class InternetClientFailureEvent extends InternetClientEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.internet.InternetClientEvent
              |
              +--org.dvb.internet.InternetClientFailureEvent
```

All Implemented Interfaces:

`java.io.Serializable`

Direct Known Subclasses:

`CancelledByUserEvent`, `PermissionDeniedEvent`, `URLUnavailableEvent`

Description

Event indicating that an operation on an internet client failed. Typically, internet clients will post subclasses of this event detailing the reason why the operation failed.

Constructors

InternetClientFailureEvent(Object, URL)

```
public InternetClientFailureEvent(java.lang.Object source, java.net.URL url)
```

Construct a new `InternetClientFailureEvent`

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

Parameters:

`source` - the source of the event.

`url` - the URL to which the event relates.

org.dvb.internet InternetClientListener

Declaration

```
public interface InternetClientListener extends java.util.EventListener
```

All Superinterfaces:

`java.util.EventListener`

Description

Interface for objects that wish to receive `InternetClientEvents`.

Methods

receiveInternetClientEvent(InternetClientEvent)

```
public void receiveInternetClientEvent(org.dvb.internet.InternetClientEvent event)
```

The method to be called when an `InternetClientEvent` is received.

Parameters:

`event` - the received `InternetClientEvent`

org.dvb.internet InternetClientService

Declaration

```
public interface InternetClientService extends javax.tv.service.Service
```

All Superinterfaces:

`javax.tv.service.Service`

All Known Subinterfaces:

`EmailClientService`, `UsenetClientService`, `WWWBrowserService`

Description

The base class for the interface to resident applications that are supported by the internet access profile.

The lifecycle of an application which implements this interface or its subclasses is for a broadcast service. The application is started by selecting the appropriate service (using the `Locator` object returned by calls to `getLocator()`). If this service is selected in the service context which contains the executing application, any currently presented content will be stopped and the application will be destroyed before the client is launched. Calling `destroy()` or `stop()` on the service context in which the client is running will cause the client to be terminated.

Methods in this API will not affect the lifecycle of the calling application.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 717 of 861

Methods

canRunApplication()

```
public boolean canRunApplication()
```

Returns true if the application can run without having to stop the downloaded GEM application.

Returns:

true if the application can be run without stopping the calling application, or false otherwise.

getName()

```
public java.lang.String getName()
```

Returns a short service name or an acronym. In the case of subclasses of `InternetClient`, the returned value is implementation dependent

Overrides:

`getName` in interface `Service`

Returns:

A string representing this service's short name.

getServiceType()

```
public javax.tv.service.ServiceType getServiceType()
```

Returns the type of this service. In the case of internet clients, one of the service types defined in the `InternetServiceType` class shall be returned.

Overrides:

`getServiceType` in interface `Service`

Returns:

The service type of this Service.

getSupportedClientServices()

```
public org.dvb.internet.InternetClientService[] getSupportedClientServices()
```

Returns all `InternetClientService`s supported by the same application as this one. This `InternetClientService` is included in the array.

Returns:

an array of `InternetClientServices`

hasMultipleInstances()

```
public boolean hasMultipleInstances()
```

This method indicates whether the service represented by this Service is available on multiple transports. This method has no effect in the case of an `InternetClient`

Overrides:

`hasMultipleInstances` in interface `Service`

Returns:

FALSE always for `InternetClient` instances.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 718 of 861

retrieveDetails(SIRequestor)

```
public javax.tv.service.SIRequest retrieveDetails(javax.tv.service.SIRequestor requestor)
```

This method will always fail when called for an `InternetClient`. The requestor will always be notified of a failure of type DATA_UNAVAILABLE.

Overrides:

`retrieveDetails` in interface `Service`

Parameters:

`requestor` - - The `SIRequestor` to be notified when this retrieval operation completes.

Returns:

An `SIRequest` object identifying the request.

org.dvb.internet

InternetClientSuccessEvent

Declaration

```
public class InternetClientSuccessEvent extends InternetClientEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.internet.InternetClientEvent
              |
              +--org.dvb.internet.InternetClientSuccessEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Event indicating that an operation on an internet client succeeded.

Constructors

InternetClientSuccessEvent(Object, URL)

```
public InternetClientSuccessEvent(java.lang.Object source, java.net.URL url)
```

Construct a new `InternetClientSuccessEvent`

Parameters:

`source` - the source of the event.

`url` - the URL to which the event relates.

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

org.dvb.internet
InternetServiceFilter

Declaration

```
public final class InternetServiceFilter extends javax.tv.service.navigation.ServiceFilter

java.lang.Object
  |
  +--javax.tv.service.navigation.ServiceFilter
        |
        +--org.dvb.internet.InternetServiceFilter
```

Description

`InternetServiceFilter` represents a service type for a particular kind of internet client. A `ServiceList` resulting from this filter will include only services providing access to the specified type of internet client.

Fields

EMAIL_CLIENT

```
public static final int EMAIL_CLIENT
```

Constant identifying an email client service.

NEWS_CLIENT

```
public static final int NEWS_CLIENT
```

Constant identifying a usenet news client service.

WWW_CLIENT

```
public static final int WWW_CLIENT
```

Constant identifying a WWW client service.

Constructors

InternetServiceFilter(int)

```
public InternetServiceFilter(int service_type)
```

Constructs the filter based on a particular type of internet client service. The types of service required are those defined by the constants in this class. Support for other values is platform dependent. Platforms not supporting services of the type specified shall return an empty `ServiceList` when instances of this class constructed using that type are used.

Parameters:

`service_type` - the type of service required.

Methods

accept(Service)

```
public boolean accept(javax.tv.service.Service service)
```

Tests if a particular service represents an internet client of the type specified in the constructor of this instance.

Overrides:

`accept` in class `ServiceFilter`

Parameters:

`service` - A Service to be evaluated against the filtering algorithm.

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

Returns:

true if service satisfies the filtering algorithm; false otherwise.

org.dvb.internet

InternetServiceType

Declaration

```
public class InternetServiceType extends javax.tv.service.ServiceType
```

```
java.lang.Object
  |
  +--javax.tv.service.ServiceType
        |
        +--org.dvb.internet.InternetServiceType
```

Description

Class representing the additional service types available in the Internet Access profile. When this service type is used in a `java.tv.service.navigation.ServiceTypeFilter` for filtering available services on their type, all internet client services shall be returned. Applications wishing to obtain a specific sub-type of internet client service should use `InternetServiceFilter`.

See Also:

`InternetServiceFilter`

Fields

INTERNET_CLIENT

```
public static final javax.tv.service.ServiceType INTERNET_CLIENT
```

WWW service type.

Constructors

InternetServiceType(String)

```
protected InternetServiceType(java.lang.String name)
```

This protected constructor is provided for implementation use and to enable future evolution of this or other specifications. It shall not be called by inter-operable applications.

Parameters:

`name` - the name of the service type.

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

org.dvb.internet PermissionDeniedEvent

Declaration

```
public class PermissionDeniedEvent extends InternetClientFailureEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.internet.InternetClientEvent
              |
              +--org.dvb.internet.InternetClientFailureEvent
                    |
                    +--org.dvb.internet.PermissionDeniedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Event indicating that an operation on an internet client failed due to the specified URL not being accessible by the client due to access controls on the server.

Constructors

PermissionDeniedEvent(Object, URL)

```
public PermissionDeniedEvent(java.lang.Object source, java.net.URL url)
```

Construct a new `PermissionDeniedEvent`.

Parameters:

`source` - the source of the event.

`url` - the URL to which the event relates.

org.dvb.internet URLUnavailableEvent

Declaration

```
public class URLUnavailableEvent extends InternetClientFailureEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.internet.InternetClientEvent
              |
              +--org.dvb.internet.InternetClientFailureEvent
                    |
                    +--org.dvb.internet.URLUnavailableEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```


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

Description

Event indicating that an operation on an internet client failed due to the specified URL not being available. This could be because a server is not available or because a file is not available on that server.

Constructors

URLUnavailableEvent(Object, URL)

```
public URLUnavailableEvent(java.lang.Object source, java.net.URL url)
```

Construct a new `URLUnavailableEvent`.

Parameters:

`source` - the source of the event.

`url` - the URL to which the event relates.

org.dvb.internet

UsenetClient

Declaration

```
public interface UsenetClient extends InternetClient
```

All Superinterfaces:

`InternetClient`, `javax.tv.service.selection.ServiceContentHandler`

Description

This interface supports the operations required on a Usenet news client.

Any URLs passed to methods in this interface should correspond to the usenet news URL format specified in RFC 1738 [61].

Methods

selectGroup(URL)

```
public void selectGroup(java.net.URL group)
throws ClientNotRunningException
```

Select and display messages in the specified newsgroup. If the news client is not running, then this method will not cause it to be started and the call will fail.

This is an asynchronous operation, whose success or failure will be indicated by an `InternetClientSuccessEvent` or `InternetClientFailureEvent` or one of their subclasses.

Parameters:

`group` - the URL of the group. This may or may not include the address of a news server.

Throws:

`java.lang.SecurityException` - if the caller does not have a `SocketPermission` for the host part of the specified URL.

`java.lang.IllegalArgumentException` - if the specified URL does not correspond to the Usenet news URL format specified in RFC 1738 [61].

`ClientNotRunningException` - if the client is not currently running.

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

selectMessage(URL)

```
public void selectMessage(java.net.URL message)
throws ClientNotRunningException
```

Select and display a message with the given message ID. If the news client is not running, then this method will not cause it to be started and the call will fail.

This is an asynchronous operation, whose success or failure will be indicated by an `InternetClientSuccessEvent` or `InternetClientFailureEvent` or one of their subclasses.

Parameters:

`message` - the URL of the message. This may or may not include the address of a news server.

Throws:

`java.lang.SecurityException` - if the caller does not have a `SocketPermission` for the host part of the specified URL

`java.lang.IllegalArgumentException` - if the specified URL does not include a Usenet news message ID or does not correspond to the Usenet news URL format specified in RFC 1738 [61].

`ClientNotRunningException` - if the client is not currently running.

org.dvb.internet

UsenetClientService

Declaration

```
public interface UsenetClientService extends InternetClientService
```

All Superinterfaces:

`InternetClientService`, `javax.tv.service.Service`

Description

Service representing the resident usenet news client.

Methods

setInitialGroup(URL)

```
public void setInitialGroup(java.net.URL group)
```

Set the initial group to be displayed when the news client starts. This URL is specific to this instance of UsenetClient-Service and will not impact any other instance and is only valid for the lifetime of this instance. Calling this method and then selecting the UsenetClientService instance is equivalent to selecting the UsenetClientService instance, obtaining the UsenetClient and then calling the selectMessage method there. If the application calling this method is still running when the message is sent (or fails) and has registered an InternetClientListener then the appropriate InternetClientEvent shall be sent corresponding to the success or failure of the operation to send the message. Calls to setInitialGroup shall cancel previous calls to setInitialMessage and vice-versa on the same UsenetClientService instance.

Parameters:

`group` - the URL of the message. This may or may not include the address of a news server.

Throws:

`java.lang.IllegalArgumentException` - if the specified URL does not correspond to the Usenet news URL format specified in RFC 1738 [61].

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

setInitialMessage(URL)

```
public void setInitialMessage(java.net.URL message)
```

Set the initial message to be used when the news client starts. This URL is specific to this instance of UsenetClient-Service and will not impact any other instance and is only valid for the lifetime of this instance. Calling this method and then selecting the UsenetClientService instance is equivalent to selecting the UsenetClientService instance, obtaining the UsenetClient and then calling the selectMessage method there. If the application calling this method is still running when the message is sent (or fails) and has registered an InternetClientListener then the appropriate InternetClientEvent shall be sent corresponding to the success or failure of the operation to send the message. Calls to setInitialGroup shall cancel previous calls to setInitialMessage and vice-versa on the same UsenetClientService instance.

Parameters:

`message` - the URL of the message. This may or may not include the address of a news server.

Throws:

`java.lang.IllegalArgumentException` - if the specified URL does not include a Usenet news message ID or does not correspond to the Usenet news URL format specified in RFC 1738 [61].

subscribe(String)

```
public void subscribe(java.lang.String newsgroup)
throws IOException
```

Add a newsgroup to the list currently subscribed newsgroups. If the newsgroup is already in the list, then this method has no effect. As a side-effect a newsgroup previously subscribed to by this method may be unsubscribed. Implementations may restrict the number of newsgroups a single GEM application or source of applications may subscribe to.

Parameters:

`newsgroup` - the name of the newsgroup that should be subscribed to.

Throws:

IOException - if no more newsgroups can be added due to a lack of storage space.

`java.io.IOException`.

org.dvb.internet

WWWBrowser

Declaration

```
public interface WWWBrowser extends InternetClient
```

All Superinterfaces:

```
InternetClient, javax.tv.service.selection.ServiceContentHandler
```

Description

This interface provides support for the operations required on an WWW browser.

Any URLs passed to methods in this interface should correspond to the HTTP or FTP URL schemes specified in RFC 1738 [61]. Other schemes or URL formats may be supported, but these are implementation-specific and additional supported schemes should be discovered by querying the capabilities of the web browser if they are required.

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

Methods

goToURL(URL)

```
public void goToURL(java.net.URL url)
throws ClientNotRunningException
```

Direct the web browser to display the page at the specified URL. If the web browser is not already running, then this method will not cause the web browser to be started and the call will fail.

This is an asynchronous operation, whose success or failure will be indicated by an `InternetClientSuccessEvent` or `InternetClientFailureEvent` or one of their subclasses.

Parameters:

`url` - the URL to visit.

Throws:

`java.lang.SecurityException` - if the caller does not have a `SocketPermission` for the host part of the specified URL.

`java.lang.IllegalArgumentException` - if the URL scheme is not supported by the web browser.

`ClientNotRunningException` - if the client is not currently running.

org.dvb.internet
WWWBrowserService

Declaration

```
public interface WWWBrowserService extends InternetClientService
```

All Superinterfaces:

`InternetClientService`, `javax.tv.service.Service`

Description

Service representing a resident WWW browser.

Methods

addBookmark(URL, String)

```
public void addBookmark(java.net.URL bookmarkUrl, java.lang.String name)
throws EntryExistsException, IOException
```

Add a bookmark to the list of bookmarks in the current application. As a side-effect a bookmark previously added by this method may be lost. Implementations may restrict the number of bookmarks a single GEM application or source of applications may add.

Parameters:

`bookmarkUrl` - the URL that should be added to the bookmarks list.

`name` - the name that should be displayed for that URL in the bookmarks list.

Throws:

`EntryExistsException` - if a bookmark with both the same name and the same URL already exists in the bookmarks list.

`java.lang.IllegalArgumentException` - if the URL scheme is not supported by the application.

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

IOException - if no more bookmarks can be added due to a lack of storage space or other limitation in the client.

`java.io.IOException`.

areFramesSupported()

```
public boolean areFramesSupported()
```

Check whether frames are supported by the browser and enabled.

Returns:

true if the browser supports frames and frame support is enabled by the user, false otherwise.

getAcceptedMediaTypes()

```
public java.lang.String[] getAcceptedMediaTypes()
```

Returns an array of supported MIME types, e.g. "application/vnd.rn-realmedia" The returned MIME types shall include at least one entry with type "text/html". Each entry of this type shall have a 'version' parameter that indicates the version of HTML that is supported, for example text/html; version = "4.0". A browser may claim to support an HTML version while only implementing a subset of the features - this is implementation-dependent. A browser that wishes to explicitly indicate support for multiple HTML versions shall return multiple entries of type "text/html" with different values for the 'version' parameter.

Returns:

an array of MIME types supported by the web browser.

getSupportedPlugins()

```
public java.lang.String[] getSupportedPlugins()
```

Returns an array of names of installed plug-ins. The name in each string is defined by the plug-in provider.

Returns:

an array of plugin names.

getUserAgent()

```
public java.lang.String getUserAgent()
```

Returns the string used in the HTTP "User-Agent" header.

Returns:

the string identifying the user agent.

setHomepage(URL)

```
public void setHomepage(java.net.URL defaultUrl)
```

Set the home page for the web browser.

Parameters:

`defaultUrl` - the URL to be used as the default when the application is launched with no starting URL.

Throws:

`java.lang.SecurityException` - if the caller does not have a `HomePagePermission`.

`java.lang.IllegalArgumentException` - if the URL scheme is not supported by the application. Different classes which implement or extend this interface may implement different schemes.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 727 of 861

setInitialURL(URL)

```
public void setInitialURL(java.net.URL initialUrl)
```

Set the initial URL to be used when the WWW browser starts. This URL is specific to this instance of WWWBrowserService and will not impact any other instance and is only valid for the lifetime of this instance. Calling this method and then selecting the WWWBrowserService instance is equivalent to selecting the WWWBrowserService instance, obtaining the WWWBrowser and then calling the goToURL method there. If the application calling this method is still running when the specified initial page is displayed (or fails) and has registered an InternetClientListener then the appropriate InternetClientEvent shall be sent corresponding to the success or failure of the operation to display the specified initial URL.

Parameters:

`initialUrl` - the URL to use.

Throws:

`java.lang.IllegalArgumentException` - if the URL scheme is not supported by the application. Different classes which implement or extend this interface may implement different schemes.

`java.lang.SecurityException` - if the caller does not have a `SocketPermission` for the host part of the specified URL.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 728 of 861

Annex AI (normative):
DVB Extensions for cryptography

Package

org.dvb.security

Description

Enables applications to login to a PKCS11 token for non key related operations including providing PIN codes.

The AuthProvider class is needed to log into a PKCS11 token for non key related operations. In that case, there is no other way to send the PIN code. The KeyStoreBuilder class is needed to install a callback handler which is called when a PIN code is required to create a KeyStore. This is an alternative to providing the PIN code when a KeyStore is instantiated. It enables an application to log into the token for key operations without using a KeyStoreBuilder.

Class Summary	
Interfaces	
KeyStoreProtectionParameters	A marker interface for keystore protection parameters.
Classes	
AuthProvider	This class defines login and logout for a provider.
DVBKeyStore	Extends KeyStore to allow loading the keystore using protection parameters.
KeyStoreBuilder	An instance of this class encapsulates the information needed to instance and initialize a KeyStore object.
KeyStoreCallbackHandlerProtection	A protection parameter encapsulating a CallbackHandler.
Exceptions	
KeyStoreException	Generic KeyStore exception.
LoginException	Basic login exception.

org.dvb.security

AuthProvider

Declaration

```
public abstract class AuthProvider extends java.security.Provider
```

```
java.lang.Object
  |
  +--java.util.Dictionary
        |
        +--java.util.Hashtable
              |
              +--java.util.Properties
                    |
                    +--java.security.Provider
                          |
                          +--org.dvb.security.AuthProvider
```

All Implemented Interfaces:

```
java.lang.Cloneable, java.util.Map, java.io.Serializable
```

Direct Known Subclasses:

```
org.dvb.security.pkcs11.DVBPKCS11Provider
```


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

Description

This class defines login and logout for a provider. While callers may invoke login directly, the provider may also invoke login on behalf of callers if it determines that a login must be perfomed prior to certain operations.

Constructors

AuthProvider(String, double, String)

```
protected AuthProvider(java.lang.String name, double version, java.lang.String info)
```

Creates a new instance of AuthProvider.

Methods

login(Principal, CallbackHandler)

```
public abstract void login(java.security.Principal identity,
org.dvb.auth.callback.CallbackHandler handler)
throws LoginException
```

log in to this provider.

Throws:

LoginException

logout()

```
public abstract void logout()
throws LoginException
```

logout from this provider.

Throws:

LoginException

setCallbackHandler(CallbackHandler)

```
public abstract void setCallbackHandler(org.dvb.auth.callback.CallbackHandler handler)
```

set a call back handler.

org.dvb.security

DVBKeyStore

Declaration

```
public class DVBKeyStore extends java.security.KeyStore
```

```
java.lang.Object
  |
  +--java.security.KeyStore
        |
        +--org.dvb.security.DVBKeyStore
```

Description

Extends KeyStore to allow loading the keystore using protection parameters.

Since:

MHP 1.1.2.

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

Constructors

DVBKeyStore(KeyStoreSpi, Provider, String)

```
protected DVBKeyStore(java.security.KeyStoreSpi keyStoreSpi, java.security.Provider provider,
java.lang.String type)
```

Creates a KeyStore object of the given type, and encapsulates the given provider implementation (SPI object) in it.

Parameters:

`keyStoreSpi` - the provider implementation.

`provider` - the provider.

`type` - the keystore type.

Methods

load(KeyStoreProtectionParameters)

```
public final void load(org.dvb.security.KeyStoreProtectionParameters p)
throws IOException, NoSuchAlgorithmException, CertificateException
```

Loads this keystore using the given protection parameters.

Parameters:

`p` - protection parameters to use.

Throws:

`java.lang.IllegalArgumentException` - if the parameter p is not recognized.

`java.io.IOException` - if there is an I/O or format problem with the keystore data.

`java.security.NoSuchAlgorithmException` - if the algorithm used to check the integrity of the keystore cannot be found.

`java.security.cert.CertificateException` - if any of the certificates in the keystore could not be loaded.

org.dvb.security

KeyStoreBuilder

Declaration

```
public abstract class KeyStoreBuilder

java.lang.Object
  |
  +--org.dvb.security.KeyStoreBuilder
```

Description

An instance of this class encapsulates the information needed to instance and initialize a KeyStore object. That process is triggered when the getKeyStore method is called. This makes it possible to decouple configuration from KeyStore object creation and delay password prompt until it is needed.

Constructors

KeyStoreBuilder()

```
protected KeyStoreBuilder()
```

Creates a new instance of KeyStoreBuilder.

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

Methods

getKeyStore()

```
public abstract java.security.KeyStore getKeyStore()
throws KeyStoreException
```

Returns the KeyStore described by this object.

Throws:

KeyStoreException - if an error occurred, e.g. if an error occurred in the constructor or the load method of the KeyStore.

newInstance(String, Provider, KeyStoreProtectionParameters)

```
public static org.dvb.security.KeyStoreBuilder newInstance(java.lang.String type,
java.security.Provider provider, org.dvb.security.KeyStoreProtectionParameters protection)
```

Returns a new builder object. Each call to the `getKeyStore` method on the returned builder will return a new org.dvb.security.DVBKeyStore object of type type. Its `load` method is invoked with the protection parameter used to construct this KeyStoreBuilder.

Parameters:

`type` - the type of the KeyStore to be constructed. The type parameter is concatenated with the string "KeyStore." and then passed to the get method of the specified Provider in order to obtain the fully qualified name of the KeyStoreSpi implementation. For more details, see "How to Implement a Provider for the JavaTM Cryptography Architecture".

`provider` - the provider from which the keyStore is to be instantiated.

`protection` - the protection parameter securing the Keystore.

Throws:

`java.lang.IllegalArgumentException` - if protection is an application defined class.

`java.lang.NullPointerException` - if type, provider or protection are null.

org.dvb.security

KeyStoreCallbackHandlerProtection

Declaration

```
public class KeyStoreCallbackHandlerProtection implements KeyStoreProtectionParameters
```

```
java.lang.Object
  |
  +--org.dvb.security.KeyStoreCallbackHandlerProtection
```

All Implemented Interfaces:

KeyStoreProtectionParameters

Description

A protection parameter encapsulating a CallbackHandler.

Constructors

KeyStoreCallbackHandlerProtection(CallbackHandler)

```
public KeyStoreCallbackHandlerProtection(org.dvb.auth.callback.CallbackHandler handler)
```

Creates a new instance of KeyStoreCallbackHandlerProtection.

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

Methods

getCallbackHandler()

```
public org.dvb.auth.callback.CallbackHandler getCallbackHandler()
```

Returns the callback handler.

org.dvb.security KeyStoreException

Declaration

```
public class KeyStoreException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.security.KeyStoreException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Generic KeyStore exception.

Constructors

KeyStoreException()

```
public KeyStoreException()
```

Creates a new instance of `KeyStoreException` without detail message.

KeyStoreException(String)

```
public KeyStoreException(java.lang.String msg)
```

Constructs an instance of `KeyStoreException` with the specified detail message.

Parameters:

`msg` - the detail message.

org.dvb.security KeyStoreProtectionParameters

Declaration

```
public interface KeyStoreProtectionParameters
```

All Known Implementing Classes:

`KeyStoreCallbackHandlerProtection`

Description

A marker interface for keystore protection parameters.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 733 of 861

org.dvb.security

LoginException

Declaration

```
public class LoginException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.security.LoginException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Basic login exception.

Constructors

LoginException()

```
public LoginException()
```

Creates a new instance of `LoginException` without detail message.

LoginException(String)

```
public LoginException(java.lang.String msg)
```

Constructs an instance of `LoginException` with the specified detail message.

Parameters:

`msg` - the detail message.

Package

org.dvb.auth.callback

Description

Enables applications to ask the end-user for PIN codes. 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.

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

Class Summary	
Interfaces	
Callback	Implementations of this interface are passed to a CallbackHandler, allowing underlying security services the ability to interact with a calling application to retrieve specific authentication data such as usernames and passwords, or to display certain information, such as error and warning messages.
CallbackHandler	An application implements a CallbackHandler and passes it to underlying security services so that they may interact with the application to retrieve specific authentication data, such as usernames and passwords, or to display certain information, such as error and warning messages.
Classes	
PasswordCallback	Underlying security services instantiate and pass a PasswordCallback to the handle method of a CallbackHandler to retrieve password information.
Exceptions	
UnsupportedCallbackException	Thrown when a callback handler is asked to handle a callback which it cannot handle.

org.dvb.auth.callback

Callback

Declaration

```
public interface Callback
```

All Known Implementing Classes:

PasswordCallback

Description

Implementations of this interface are passed to a CallbackHandler, allowing underlying security services the ability to interact with a calling application to retrieve specific authentication data such as usernames and passwords, or to display certain information, such as error and warning messages. Callback implementations do not retrieve or display the information requested by underlying security services. Callback implementations simply provide the means to pass such requests to applications, and for applications, if appropriate, to return requested information back to the underlying security services.

org.dvb.auth.callback

CallbackHandler

Declaration

```
public interface CallbackHandler
```

Description

An application implements a CallbackHandler and passes it to underlying security services so that they may interact with the application to retrieve specific authentication data, such as usernames and passwords, or to display certain information, such as error and warning messages.

Methods

handle(Callback[])

```
public void handle(org.dvb.auth.callback.Callback[] callbacks)
throws IOException, UnsupportedCallbackException
```

Called when a callback needs handling.

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

Parameters:

callbacks - - an array of Callback objects provided by an underlying security service which contains the information requested to be retrieved or displayed.

Throws:

java.io.IOException - if an input or output error occurs when retrieving the password.

UnsupportedCallbackException - if one of the callbacks in the array is not supported by the handler.

org.dvb.auth.callback

PasswordCallback

Declaration

```
public class PasswordCallback implements Callback
```

```
java.lang.Object
  |
  +--org.dvb.auth.callback.PasswordCallback
```

All Implemented Interfaces:

Callback

Description

Underlying security services instantiate and pass a PasswordCallback to the handle method of a CallbackHandler to retrieve password information. The CallbackHandler uses this to communicate to the security services a password obtained from the end-user.

Constructors

PasswordCallback(String, boolean)

```
public PasswordCallback(java.lang.String prompt, boolean echoOn)
```

Creates a new instance of PasswordCallback.

Parameters:

prompt - the prompt to use.

echoOn - true if the password should be displayed as typed otherwise false.

Methods

clearPassord()

```
public void clearPassord()
```

Clear the retrieved password.

getPassword()

```
public char[] getPassword()
```

Get the retrieved password.

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

Returns:

the last password previously set by setPassword or null if none has been set.

See Also:

setPassword(char[])

getPrompt()

```
public java.lang.String getPrompt()
```

Get the prompt to use.

Returns:

the prompt as passed to the constructor.

isEchoOn()

```
public boolean isEchoOn()
```

Return whether the password should be displayed as being typed.

Returns:

true if the password should be displayed otherwise false.

setPassword(char[])

```
public void setPassword(char[] password)
```

Set the retrieved password.

Parameters:

password - the password to return.

See Also:

getPassword()

org.dvb.auth.callback UnsupportedCallbackException

Declaration

```
public class UnsupportedCallbackException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.auth.callback.UnsupportedCallbackException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown when a callback handler is asked to handle a callback which it cannot handle.

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

Constructors

UnsupportedCallbackException(Callback)

```
public UnsupportedCallbackException(org.dvb.auth.callback.Callback callback)
```

Creates a new instance of `UnsupportedCallbackException` without detail message.

Parameters:

`callback` - the callback which cannot be handled.

UnsupportedCallbackException(Callback, String)

```
public UnsupportedCallbackException(org.dvb.auth.callback.Callback callback, java.lang.String msg)
```

Constructs an instance of `UnsupportedCallbackException` with the specified detail message.

Parameters:

`callback` - the callback which cannot be handled.

`msg` - the detail message.

Methods

getCallback()

```
public org.dvb.auth.callback.Callback getCallback()
```

Returns the Callback passed in to the constructor.

Returns:

a callback.

Package org.dvb.net.ssl

Description

Enables applications to provide keys and certificates for SSL/TLS connections. 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.

The init method in DVBTrustManagerFactory and DVBKeyManagerFactory 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.

Class Summary	
Classes	
DVBKeyManagerFactory	This class is used to create a KeyManagerFactory initialized with an array of KeyStoreBuilder instances.
DVBKeyManagerFactorySpi	This class defines the Service Provider Interface for the DVBKeyManager-Factory class.
DVBTrustManagerFactory	This class is used to create a KeyManagerFactory initialized with an array of KeyStoreBuilder instances.
DVBTrustManagerFactorySpi	This class defines the Service Provider Interface for the DVBTrustManager-Factory class.

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

org.dvb.net.ssl

DVBKeyManagerFactory

Declaration

```
public class DVBKeyManagerFactory extends javax.net.ssl.KeyManagerFactory

java.lang.Object
  |
  +--javax.net.ssl.KeyManagerFactory
        |
        +--org.dvb.net.ssl.DVBKeyManagerFactory
```

Description

This class is used to create a KeyManagerFactory initialized with an array of KeyStoreBuilder instances. The (inherited) method getInstance shall return an instance of DVBKeyManagerFactory where provider is an instance of DVBPKCS11Provider.

Constructors

DVBKeyManagerFactory(DVBKeyManagerFactorySpi, Provider, String)

```
protected DVBKeyManagerFactory(org.dvb.net.ssl.DVBKeyManagerFactorySpi factorySpi,
java.security.Provider provider, java.lang.String algorithm)
```

Creates a new instance of DVBKeyManagerFactory.

Methods

init(KeyStoreBuilder[])

```
public final void init(org.dvb.security.KeyStoreBuilder[] builders)
```

This method is used to initialize the factory with an array of KeyStoreBuilder instances.

Parameters:

`builders` - an array of KeyStoreBuilder instances.

org.dvb.net.ssl

DVBKeyManagerFactorySpi

Declaration

```
public abstract class DVBKeyManagerFactorySpi extends javax.net.ssl.KeyManagerFactorySpi

java.lang.Object
  |
  +--javax.net.ssl.KeyManagerFactorySpi
        |
        +--org.dvb.net.ssl.DVBKeyManagerFactorySpi
```

Description

This class defines the Service Provider Interface for the DVBKeyManagerFactory class.

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

Constructors

DVBKeyManagerFactorySpi()

```
public DVBKeyManagerFactorySpi()
```

Creates a new instance of DVBKeyManagerFactorySpi.

Methods

engineInit(KeyStoreBuilder[])

```
protected abstract void engineInit(org.dvb.security.KeyStoreBuilder[] builders)
```

This method is used to initialize the factory with an array of KeyStoreBuilder instances.

Parameters:

`builders` - an array of KeyStoreBuilder.

org.dvb.net.ssl

DVBTrustManagerFactory

Declaration

```
public class DVBTrustManagerFactory extends javax.net.ssl.TrustManagerFactory
```

```
java.lang.Object
  |
  +--javax.net.ssl.TrustManagerFactory
        |
        +--org.dvb.net.ssl.DVBTrustManagerFactory
```

Description

This class is used to create a KeyManagerFactory initialized with an array of KeyStoreBuilder instances. The inherited method getInstance shall return an instance of DVBTrustManagerFactory when the provider is an instance of DVBPKCS11Provider.

Constructors

DVBTrustManagerFactory(DVBTrustManagerFactorySpi, Provider, String)

```
protected DVBTrustManagerFactory(org.dvb.net.ssl.DVBTrustManagerFactorySpi factorySpi,
java.security.Provider provider, java.lang.String algorithm)
```

Creates a new instance of DVBTrustManagerFactory.

Methods

init(KeyStoreBuilder[])

```
public final void init(org.dvb.security.KeyStoreBuilder[] builders)
```

This method is used to initialize the factory with an array of KeyStoreBuilder instances.

Parameters:

`builders` - an array of KeyStoreBuilder instances.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 740 of 861

org.dvb.net.ssl DVBTrustManagerFactorySpi

Declaration

```
public abstract class DVBTrustManagerFactorySpi extends javax.net.ssl.TrustManagerFactorySpi

java.lang.Object
  |
  +--javax.net.ssl.TrustManagerFactorySpi
        |
        +--org.dvb.net.ssl.DVBTrustManagerFactorySpi
```

Description

This class defines the Service Provider Interface for the DVBTrustManagerFactory class.

Constructors

DVBTrustManagerFactorySpi()

```
public DVBTrustManagerFactorySpi()
```

Creates a new instance of DVBTrustManagerFactorySpi.

Methods

engineInit(KeyStoreBuilder[])

```
protected abstract void engineInit(org.dvb.security.KeyStoreBuilder[] builders)
```

This method is used to initialize the factory with an array of KeyStoreBuilder instances.

Parameters:

`builders` - an array of `KeyStoreBuilder`s.

Package org.dvb.security.pkcs11

Description

Provides information about available slots and tokens and enables the selection of the slot to use.

PKCS11 smart card products in the market show significant variation. In practice, this variation is significant enough that a defining a default PKCS11 driver to be resident in all GEM terminals is impractical. DVBPKCS11Provider implementations will use the " Non-CA smart card API" (http://java.sun.com/products/satsa/) to communicate with the smart cards.

In this package, when PKCS#11 library functions are mentioned, it is to indicate that the call this method shall cause effect similar to the named PKCS#11 library equivalent.

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.

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

Class Summary	
Interfaces	
SlotInfo	This interface is used to retrieve information about a PKCS11 slot.
TokenInfo	This interface is used to get information about a PKCS11 token.
Classes	
DVBPKCS11Provider	This class implements a PKCS11 security provider.

org.dvb.security.pkcs11

DVBPKCS11Provider

Declaration

```
public abstract class DVBPKCS11Provider extends org.dvb.security.AuthProvider
```

```
java.lang.Object
  |
  +--java.util.Dictionary
        |
        +--java.util.Hashtable
              |
              +--java.util.Properties
                    |
                    +--java.security.Provider
                          |
                          +--org.dvb.security.AuthProvider
                                |
                                +--org.dvb.security.pkcs11.DVBPKCS11Provider
```

All Implemented Interfaces:

```
java.lang.Cloneable, java.util.Map, java.io.Serializable
```

Description

This class implements a PKCS11 security provider. It can be used in the following security related packages:

- In java.security for MessageDigest, SecureRandom, Signature, KeyPairGenerator and KeyStore related classes.
- In javax.crypto (JCE) for encryption and decryption.

Providers have a slot identifier associated with them identifying each smart card reader slot. These are numbered starting from zero. For details, see the PKCS 11 specification.

Constructors

DVBPKCS11Provider(String, double, String, AppID)

```
protected DVBPKCS11Provider(java.lang.String name, double version, java.lang.String info,
org.dvb.application.AppID appID)
throws SecurityException
```

Creates a new instance of DVBPKCS11Provider.

Throws:

`java.lang.SecurityException` - when called with an appID not allowed by the DVBPKCS11 implementation.

Methods

getSlotId()

```
public abstract int getSlotId()
```


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

This method returns the PKCS11 slot identifier currently associated with this provider. By default the slot identifier is defined by the property "dvb.security.pkcs11.defaultSlotId" It can be changed by calling setSlotId.

Returns:

a slot identifier.

getSlotList(boolean)

```
public org.dvb.security.pkcs11.SlotInfo[] getSlotList(boolean tokenPresent)
```

This method is used to get the list of PKCS11 Slot available for this provider. It is equivalent to the PKCS11 C_GetSlotList function to get the list of slot and C_GetSlotInfo to retreive slot information.

Parameters:

`tokenPresent` - boolean indicating if the returned list includes only the slots with a token present or all slots.

Returns:

the list of slot.

getTokenInfo(int)

```
public abstract org.dvb.security.pkcs11.TokenInfo getTokenInfo(int slotId)
throws IllegalArgumentException
```

This method is used to retreive information about a PKCS11 token in a given slot. It is equivalent to the PKCS11 C_GetTokenInfo function to get this information.

Returns:

a TokenInfo object which gives a subset of the PKCS11 CK_TOKEN_INFO.

Throws:

`java.lang.IllegalArgumentException` - if the slot does not exist or there is no token in the slot.

login(Principal, CallbackHandler)

```
public abstract void login(java.security.Principal identity,
org.dvb.auth.callback.CallbackHandler handler)
throws LoginException, NullPointerException
```

This method is used to explicitly log into a PKCS11 token. A call to this method will be equivalent to a call to the PKCS11 function C_Login. The PKCS11 user type will always be CKU_USER.

Overrides:

`login` in class `AuthProvider`

Parameters:

`identity` - This parameter is not used. It is kept to be compatible with the J2SE 5.0.

`handler` - This parameter is used to get the pin code needed to login. The callbackHandler will get a PasswordCallback in which it should put the pin code. This parameter may be null in which case the handler that was previously set by setCallbackHandler is used.

Throws:

`org.dvb.security.LoginException` - This exception is under the conditions when C_Login would return an error return an error.

`java.lang.NullPointerException` - if the CallbackHandler parameter is null and either no previous call to setCallbackHandler has occurred or the last call to that method set the handler to null.

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

logout()

```
public abstract void logout()
throws LoginException
```

This method is called to explicitly log out from a PKCS11 token. A call to this method will be equivalent to a call to the PKCS11 funtion C_Logout.

Overrides:

logout in class AuthProvider

Throws:

org.dvb.security.LoginException - This exception is thrown under the conditions when C_Logout would return an error.

setCallbackHandler(CallbackHandler)

```
public abstract void setCallbackHandler(org.dvb.auth.callback.CallbackHandler handler)
```

This method is used to set a default callback handler for the provider.

Overrides:

setCallbackHandler in class AuthProvider

Parameters:

handler - a Callback handler that will be used to get the pin code when the login method is called with a null handler.

setSlotId(int)

```
public abstract void setSlotId(int slotId)
throws IOException, IllegalArgumentException
```

This method can be used to change the slot identifier used by the provider. The slot can only be changed when the provider is not logged into the token.

Parameters:

slotId - a slot identifier.

Throws:

java.io.IOException - this exception is thrown if this method is called when the provider is logged into the token.

java.lang.IllegalArgumentException - if the slot does not exist.

org.dvb.security.pkcs11

SlotInfo

Declaration

```
public interface SlotInfo
```

Description

This interface is used to retrieve information about a PKCS11 slot. The information returned here is the equivalent to the one returned by the PKCS11 function C_GetSlotInfo.

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

Methods

getFlags()

```
public int getFlags()
```

This method returns the flags of the PKCS11 CK_SLOT_INFO.

getManufacturerID()

```
public java.lang.String getManufacturerID()
```

This method returns the ManufacturerID of the PKCS11 CK_SLOT_INFO.

getSlotDescription()

```
public java.lang.String getSlotDescription()
```

This method returns the slotDescription of the PKCS11 CK_SLOT_INFO.

getSlotID()

```
public int getSlotID()
```

This method returns the slot identifier associated this slot information.

org.dvb.security.pkcs11

TokenInfo

Declaration

```
public interface TokenInfo
```

Description

This interface is used to get information about a PKCS11 token. The information returned here is the equivalent to the one returned by the PKCS11 function C_GetTokenInfo.

Methods

getFlags()

```
public int getFlags()
```

This method returns the flags of the PKCS11 CK_TOKEN_INFO.

getLabel()

```
public java.lang.String getLabel()
```

This method returns the label of the PKCS11 CK_TOKEN_INFO.

getManufacturerID()

```
public java.lang.String getManufacturerID()
```

This method returns the manufacturerID of the PKCS11 CK_TOKEN_INFO.

getModel()

```
public java.lang.String getModel()
```

This method returns the model of the PKCS11 CK_TOKEN_INFO.

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

getSerialNumber()

```
public java.lang.String getSerialNumber()
```

This method returns the serial Number of the PKCS11 CK_TOKEN_INFO.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 746 of 861

Annex AJ (normative): Cryptographics service provider installation

AJ.1 Introduction (informative)

PKCS11 smart card products in the market show significant variation. In practice, this variation is significant enough that a defining a default PKCS11 driver to be resident in all GEM terminals is impractical. GEM defines a mechanism to permit the downloading of cryptographic service providers in the form of Java classes. Specifically, a cryptographic service provider is a set of Java classes that provide a concrete implementation of security related functions.

To implement a cryptographic provider for the smart card, the following is needed:

a) Implement the subclasses of the needed SPI classes.
The following SPI (Service Provider Interface) classes are available :

```
java.security.SignatureSpi,
java.security.MessageDigestSpi,
java.security.KeyPairGeneratorSpi,
java.security.SecureRandomSpi,
java.security.AlgorithmParameterGeneratorSpi,
java.security.AlgorithmParametersSpi,
java.security.KeyFactorySpi,
java.security.cert.CertificateFactorySpi,
java.security.KeyStoreSpi,
javax.crypto.CipherSpi,
javax.crypto.MacSpi,
org.dvb.net.ssl.DVBKeyManagerFactorySpi,
org.dvb.net.ssl.DVBTrustManagerFactorySpi,
```

To access the smart card, the provider implementation must use the Non-CA smart card API as defined in clause 11.9.4, "Non-CA smart card API".

b) Implement the master provider class which is a subclass of org.dvb.security.pkcs11.DVBPKCS11Provider. The main purpose of this class is to give the list of the SPI classes implemented by the provider. This list is given via a set of properties initialized in the constructor of the class.

The present document does not define a mechanism by which applications may request instances of java.security.SecurityPermission. Hence applications will not be able to use the methods addProvider and removeProvider from the class java.security.Security to install and remove providers.

AJ.2 The org.dvb.security.provider package

Package

org.dvb.security.provider

Description

Enables the use of providers (as supported by the org.dvb.spi package) as cryptographic service providers (as supported by the java.security package).

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

Class Summary	
Interfaces	
CryptographicServiceProviderProvider	This xlet-bound provider permits a cryptographic service provider to be packaged and managed as part of the GEM provider framework as supported through the org.dvb.spi package.
Classes	
ProviderManager	This class gives access to those cryptographic service providers visible to an application.

org.dvb.security.provider CryptographicServiceProviderProvider

Declaration

```
public interface CryptographicServiceProviderProvider extends org.dvb.spi.XletBoundProvider
```

All Superinterfaces:

```
org.dvb.spi.Provider, org.dvb.spi.XletBoundProvider
```

Description

This xlet-bound provider permits a cryptographic service provider to be packaged and managed as part of the GEM provider framework as supported through the org.dvb.spi package. Xlet-bound providers shall not be visible to applications through the methods of the java.security.Security class.

Methods

getProvider()

```
public org.dvb.security.pkcs11.DVBPKCS11Provider getProvider()
```

Obtain a DVBPKCS11Provider to be used by this Xlet.

Returns:

a DVBPKCS11Provider.

org.dvb.security.provider ProviderManager

Declaration

```
public class ProviderManager
```

```
java.lang.Object
  |
  +--org.dvb.security.provider.ProviderManager
```

Description

This class gives access to those cryptographic service providers visible to an application. The cryptographic service providers visible to an application shall include the following:

- The installed providers as returned by java.security.Security.getProviders.
- Xlet bound providers which are bound to the calling Xlet.

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

Where providers of the same name are visible to an application through both the above mechanisms, the Xlet bound provider shall always be used by this class, even if the installed provider has a higher version number. Installed providers can always be accessed through java.security.Security.getProviders.

Constructors

ProviderManager()

```
protected ProviderManager()
```

Creates a new instance of ProviderManager. 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.

Methods

getInstance()

```
public static org.dvb.security.provider.ProviderManager getInstance()
```

Gets the singleton instance of the ProviderManager.

Returns:

the ProviderManager.

getProvider(String)

```
public java.security.Provider getProvider(java.lang.String name)
```

Returns a visible provider with the specified name. If no provider with that name is visible to the calling application then null shall be returned.

Parameters:

`name` - the name of the provider.

Returns:

a provider or null.

getProviders()

```
public java.security.Provider[] getProviders()
```

Returns an array containing all the providers visible to the calling application.

Returns:

an array of providers.

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

Annex AK (normative): Extended service selection API

Package

org.dvb.service.selection

Description

Extensions to the Java TV service selection API.

Class Summary	
Interfaces DvbServiceContext	An extension to ServiceContext permitting the discovery of the network interface (if any) being used by a service context.

org.dvb.service.selection

DvbServiceContext

Declaration

```
public interface DvbServiceContext extends javax.tv.service.selection.ServiceContext
```

All Superinterfaces:

```
javax.tv.service.selection.ServiceContext
```

Description

An extension to ServiceContext permitting the discovery of the network interface (if any) being used by a service context. When a service is selected in a service context, the resource priority of the service context shall be initialised to the resource priority of the service where one is set. Where a service does not have a resource priority, the priority of the service context shall be initialised to 128.

Methods

getNetworkInterface()

```
public org.davic.net.tuning.NetworkInterface getNetworkInterface()
```

Return the NetworkInterface reserved by this ServiceContext. The NetworkInterface instance returned shall be .equals() to one of those returned by NetworkInterfaceManager.getNetworkInterfaces().

Returns:

a NetworkInterface object or null if this ServiceContext has no NetworkInterface reserved.

getPriority()

```
public int getPriority()
```

Get the resource priority for this service context. The priority is a number between 0 and 255.

Returns:

the resource priority for this service context.

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

Since:

MHP 1.2.

setDefaultVideoTransformation(VideoTransformation)

```
public void setDefaultVideoTransformation(org.dvb.media.VideoTransformation vt)
```

Sets the default video transformation to be applied to the new service following a service selection operation. The identity of the last application to call this method for each service context instance shall be remembered by the GEM terminal. If this is set then the presentation of the video of the new service will use the video transformation of the existing service until it is known whether that application survives. If that application survives then this transformation shall be applied to the video of the new service. If that application does not survive then the video transformation of the new service will be reset to the default for the platform and the value set by the call to this method discarded. This method shall have no effect if the ServiceMediaHandler for the service context is a component based player when service selection happens.

Parameters:

`vt` - the video default transformation or null to reset to the default for the platform.

Since:

MHP 1.1.2.

setPriority(int)

```
public void setPriority(int priority)
```

Set the resource priority for this service context. The priority is a number between 0 and 255.

Parameters:

`priority` - the resource priority for this service context.

Throws:

`java.lang.IllegalArgumentException` - if the priority is outside this specified range.

Since:

MHP 1.2.

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

Annex AL (normative): Extended content referencing API

The class `org.dvb.locator.FrequencyLocator`, defined in this annex, is optional for GEM terminals.

Package org.dvb.locator

Description

Provides Locators for various concepts not addressed in the org.davic and javax.tv packages.

Class Summary	
Classes	
FrequencyLocator	Used to reference a service that does not carry any DVB-SI and hence does not appear in the service list of a receiver.
NetworkInterfaceBoundMediaLocator	Class representing a MediaLocator combined with a network interface.

org.dvb.locator FrequencyLocator

Declaration

```
public class FrequencyLocator extends org.davic.net.Locator
```

```
java.lang.Object
  |
  +--org.davic.net.Locator
        |
        +--org.dvb.locator.FrequencyLocator
```

All Implemented Interfaces:

```
javax.tv.locator.Locator
```

Description

Used to reference a service that does not carry any DVB-SI and hence does not appear in the service list of a receiver. The external form of locators of this class is implementation specific.

Since:

MHP 1.1.2.

Constructors

FrequencyLocator(byte[], int)

```
public FrequencyLocator(byte[] delivery_system_descriptor, int program_number)
throws InvalidLocatorException
```

Constructor for referencing a service which does not carry any DVB-SI.

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

Parameters:

`delivery_system_descriptor` - one of the delivery system descriptors defined in clause 6.2.13 ("Delivery system descriptors") of the DVB-SI specification.

`program_number` - the MPEG program_number of the service within the transport stream.

Throws:

`org.davic.net.InvalidLocatorException` - if the delivery_system_descriptor parameter is incorrect - the descriptor tag does not define one of the permitted descriptors, the length of the array is not correct for the descriptor tag, any values within the descriptor are outside any range defined for them.

Methods

getDeliverySystemDescriptor()

```
public byte[] getDeliverySystemDescriptor()
```

Return the delivery system descriptor as passed into the constructor.

Returns:

a byte array containing a DVB-SI delivery system descriptor.

getProgramNumber()

```
public int getProgramNumber()
```

Return the MPEG program number as passed in to the constructor.

Returns:

an MPEG program number.

org.dvb.locator

NetworkInterfaceBoundMediaLocator

Declaration

```
public class NetworkInterfaceBoundMediaLocator extends javax.media.MediaLocator
```

```
java.lang.Object
  |
  +--javax.media.MediaLocator
        |
        +--org.dvb.locator.NetworkInterfaceBoundMediaLocator
```

Description

Class representing a MediaLocator combined with a network interface. Used by applications that want to control which network interface is used by JMF when multiple network interfaces exist which are connected to the same actual network, for example a multi-tuner PVR.

JMF players constructed using instances of this class shall only use the specified network interface. If the media referenced by that media locator is not available via that network interface, the JMF player shall behave as if the media is not available at all, even if it is in fact available by another network interface.

Since:

MHP 1.1.2.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 753 of 861

Constructors

NetworkInterfaceBoundMediaLocator(MediaLocator, NetworkInterface)

```
public NetworkInterfaceBoundMediaLocator(javax.media.MediaLocator locator,
org.davic.net.tuning.NetworkInterface ni)
```

Construct an instance of this class.

Parameters:

`locator` - the MediaLocator to be combined with a NetworkInterface.

`ni` - a NetworkInterface.

Methods

getNetworkInterface()

```
public org.davic.net.tuning.NetworkInterface getNetworkInterface()
```

Return the network interface aspect of this MediaLocator.

Returns:

the network interface passed into the constructor.

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

Annex AM (normative): Smart card reader API

Package

org.dvb.smartcard

Description

Provides access to smart card readers. This access includes information about the status of any smart card in those readers and any changes in that status.

Class Summary	
Interfaces	
SmartCardReaderListener	The listener interface to receive smart card reader status changes.
Classes	
SmartCardReader	Represents a physical smart card reader slot in the terminal.
SmartCardReaderEvent	Represents an event generated by a change in the status of a smart card reader.
SmartCardReaderManager	The smart card reader manager allow user to know the status of any slot available on the terminal.

org.dvb.smartcard

SmartCardReader

Declaration

```
public class SmartCardReader

java.lang.Object
  |
  +--org.dvb.smartcard.SmartCardReader
```

Description

Represents a physical smart card reader slot in the terminal.

Since:

MHP 1.1.3.

Constructors

SmartCardReader()

```
protected SmartCardReader()
```

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.

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

Methods

addSmartCardReaderListener(SmartCardReaderListener)

```
public void addSmartCardReaderListener(org.dvb.smartcard.SmartCardReaderListener listener)
```

Allows the specified listener to received notifications about changes in the status of the smart card reader.

Parameters:

`listener` - the SmartCardReaderListener that will receive the notifications.

getSlotId()

```
public int getSlotId()
```

Each single reader is identified on the terminal with an increasing identifier. Default identifier for single slot terminals is zero.

Returns:

int id associated to the reader.

getStatus()

```
public int getStatus()
```

Retrieves current status of the smart card reader. The constant values are the ones defined in SmartCardReaderEvent.

Returns:

int current status of the reader.

isSmartCardInserted()

```
public boolean isSmartCardInserted()
```

Allows applications to query if a smart card is inserted in the reader. True is returned also if no ATR is correctly retrieved (i.e. smart card inserted upside-down).

Returns:

boolean true if smart card is inserted in given slot.

openRawConnection()

```
public javax.microedition.apdu.APDUConnection openRawConnection()
throws IOException, ConnectionNotFoundException
```

Opens a raw APDU connection without selecting an application or managing channels. The following properties shall apply to raw connections:

- There may be only one raw connection open at the same time with the same smart card.
- The implementation shall open the connection without sending any extra commands.
- A GEM application is allowed to send Application Selection (SELECT FILE by DF NAME) and MANAGE CHANNEL commands.
- For the T=0 protocol, for case 4 and case 2 command APDUs the card may respond with 61 XX or 6C XX statuses. The implementation should not handle these special cases, it should neither send GET RESPONSE, nor resend commands. The GEM application is required to handle these cases.

Returns:

an APDUConnection with the selected smartcard.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 756 of 861

Throws:

`java.lang.SecurityException` - if the application does not have APDUPermission.

`javax.microedition.io.ConnectionNotFoundException` - if either the smartcard is not inserted or if opening a connection is not permitted (e.g. for CA cards).

`java.io.IOException` - if an APDU connection is already open or if any other I/O error occurs.

removeSmartCardReaderListener(SmartCardReaderListener)

```
public void removeSmartCardReaderListener(org.dvb.smartcard.SmartCardReaderListener listener)
```

Removes specified listener: the notifications about changes in the status of the smart card reader will no longer be forwarded to the given listener.

Parameters:

`listener` - the SmartCardReaderListener that was receiving the notifications.

org.dvb.smartcard

SmartCardReaderEvent

Declaration

```
public class SmartCardReaderEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.smartcard.SmartCardReaderEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Represents an event generated by a change in the status of a smart card reader. All available constants are defined within this class.

Since:

MHP 1.1.3.

Fields

SMART_CARD_ERROR

```
public static final int SMART_CARD_ERROR
```

Smart card is inserted into the reader, there is electrical communication with the smart card but no ATR is retrieved.

SMART_CARD_IN

```
public static final int SMART_CARD_IN
```

A smart card is inserted into the reader and an ATR is correctly retrieved. This means the insertion contact is active and the reader is also able to communicate with the smart card.

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

SMART_CARD_MUTED

```
public static final int SMART_CARD_MUTED
```

Smart card is inserted into the reader but no ATR is retrieved because no electrical communication is established with the smart card.

SMART_CARD_OUT

```
public static final int SMART_CARD_OUT
```

Nothing is inserted into the reader, meaning the insertion contact is disabled.

Constructors

SmartCardReaderEvent(Object, int)

```
public SmartCardReaderEvent(java.lang.Object source, int type)
```

Constructor for a smart card reader event notifying the slot has identified a change in its status.

Parameters:

`source` - Object the `SmartCardReader` who is generating the event.

`type` - int the event type.

Methods

getType()

```
public int getType()
```

Retrieves the type of `SmartCardReaderEvent` that has been fired. It can be either `SMART_CARD_IN`, `SMART_CARD_OUT`, `SMART_CARD_MUTED` or `SMART_CARD_ERROR`.

Returns:

int type of event.

org.dvb.smartcard

SmartCardReaderListener

Declaration

```
public interface SmartCardReaderListener
```

Description

The listener interface to receive smart card reader status changes.

Since:

MHP 1.1.3.

Methods

smartCardReaderEventReceived(SmartCardReaderEvent)

```
public void smartCardReaderEventReceived(org.dvb.smartcard.SmartCardReaderEvent event)
```

Called by the terminal when a change in the status of the smart card reader is notified.

Parameters:

`event` - SmartCardReaderEvent the event that was fired.

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

org.dvb.smartcard

SmartCardReaderManager

Declaration

```
public final class SmartCardReaderManager

java.lang.Object
  |
  +--org.dvb.smartcard.SmartCardReaderManager
```

Description

The smart card reader manager allow user to know the status of any slot available on the terminal. The manager can dispatch the change of status in any reader to applications that registered themselves for monitoring it or can synchronously return the current status of the reader. The SmartCardReaderManager is a singleton.

Since:

MHP 1.1.3.

Constructors

SmartCardReaderManager()

```
protected SmartCardReaderManager()
```

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.

Methods

getInstance()

```
public static org.dvb.smartcard.SmartCardReaderManager getInstance()
```

Used to get the unique istance of the smart card reader manager.

Returns:

SmartCardReaderManager the manager singleton instance.

getNumber()

```
public int getNumber()
```

Retrieves the number of smart card readers provided on the terminal.

Returns:

int number of card readers.

getSmartCardReaders()

```
public org.dvb.smartcard.SmartCardReader[] getSmartCardReaders()
```

Allows application to retrieve an array including all the smart card readers provided on the terminal. In case no readers are provided, an array with length zero is returned.

Returns:

SmartCardReader[] array of smart card readers.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 759 of 861

Annex AN (normative): Provider APIs

Package

org.dvb.spi

Description

This package defines a central registry for all DVB Service Provider Interface (SPI) providers. This encourages a consistent architecture for such providers. Additionally, it gives some support for debugging. For example, a list of installed providers might be useful in a crash log.

In normal operation, xlets using a provider never get a direct reference to that provider. Rather, xlets interact with the terminal through standardized APIs, such as Java TV. The service underlying those underlying APIs can be:

- an integrated part of the implementation; or
- interoperable downloaded code, either in the form of an xlet-bound or a system-bound provider.

NOTE: The Java platform class `java.security.Provider` is unrelated to the providers defined in this package and its subpackages. Additionally, `java.security.Provider` in the Java platform is typically associated with a completely different lifecycle model.

Class Summary	
Interfaces	
`Provider`	Abstract interface for all DVB providers.
`SystemBoundProvider`	Interface for providers that are registered from one Xlet, but provide system-wide services.
`XletBoundProvider`	Interface for all Xlet-bound providers.
Classes	
`ProviderPermission`	This class is for applications which wish to be able to install providers.
`ProviderRegistry`	Registry of providers.
Exceptions	
`ProviderFailedInstallationException`	Thrown when there is a problem installing a provider.

org.dvb.spi

Provider

Declaration

```
public interface Provider
```

All Known Subinterfaces:

```
org.dvb.security.provider.CryptographicServiceProviderProvider,
org.dvb.spi.ict.InteractionChannelTransportProvider, org.dvb.spi.selection.SelectionProvider,
org.dvb.spi.si.full.SIManagerProvider, org.dvb.spi.si.simple.SimpleSIProvider, SystemBoundProvider,
XletBoundProvider
```

Description

Abstract interface for all DVB providers.

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

Since:

MHP 1.1.3.

Methods

getName()

```
public java.lang.String getName()
```

Returns the name of this provider. This can be used for debugging purposes, e.g. in a crash log. The name shall be encoded as defined for the permission request file in the main body of the present document. For example "0x0000000B.EMV_PK11.VISA_REVOLVER".

Returns:

the name.

getServiceProviderInterfaces()

```
public java.lang.Class[] getServiceProviderInterfaces()
```

Gives a list of the SPI's implemented by this provider. The list shall be at least one element long, and shall contain only valid SPI interfaces defined by the terminal specification. Unknown interfaces (e.g. application-defined interfaces) shall be rejected, as documented in ProviderRegistry.

Returns:

a list of SPIs.

See Also:

```
ProviderRegistry.registerXletBound(XletBoundProvider),
ProviderRegistry.registerSystemBound(SystemBoundProvider)
```

getVersion()

```
public java.lang.String getVersion()
```

Return the version of this provider. The format of this string is not specified.

Returns:

- the version of this provider.

providerRegistered()

```
public void providerRegistered()
```

Called by the system when this provider is registered.

providerUnregistered()

```
public void providerUnregistered()
```

Called by the system when this provider is unregistered.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 761 of 861

org.dvb.spi ProviderFailedInstallationException

Declaration

```
public class ProviderFailedInstallationException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.spi.ProviderFailedInstallationException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Thrown when there is a problem installing a provider.

Since:

MHP 1.1.3.

See Also:

```
ProviderRegistry.registerSystemBound(SystemBoundProvider),
ProviderRegistry.registerXletBound(XletBoundProvider)
```

Constructors

ProviderFailedInstallationException()

```
public ProviderFailedInstallationException()
```

Constructs an exception with no reason.

ProviderFailedInstallationException(String)

```
public ProviderFailedInstallationException(java.lang.String s)
```

Constructs an exception with a reason.

Parameters:

`s` - the reason why the exception was thrown.

org.dvb.spi ProviderPermission

Declaration

```
public class ProviderPermission extends java.security.BasicPermission

java.lang.Object
  |
  +--java.security.Permission
        |
        +--java.security.BasicPermission
              |
              +--org.dvb.spi.ProviderPermission
```


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

All Implemented Interfaces:

```
java.security.Guard, java.io.Serializable
```

Description

This class is for applications which wish to be able to install providers. A ProviderPermission contains a name and an action string.

The permission name shall be either the fully qualified class name of the provider class to be installed or "*" meaning any provider. Applications may have multiple instances of this permission in order to be able to install more than one but not all providers.

The actions list shall either be "xlet" or "system". "xlet" means the right to install the provider as an xlet bound provider and "system" as a system bound provider. No checking shall be performed on whether the specified class name is consistent with the action.

Since:

MHP 1.1.3.

Constructors

ProviderPermission(String)

```
public ProviderPermission(java.lang.String name)
```

Creates a new ProviderPermission with the specified name. The name is the symbolic name of the ProviderPermission.

Parameters:

`name` - the name of the ProviderPermission or "*"

ProviderPermission(String, String)

```
public ProviderPermission(java.lang.String name, java.lang.String actions)
```

Creates a new ProviderPermission object with the specified name. The name is the symbolic name of the Provider-Permission. The actions string should be either "xlet" or "system". This constructor exists for use by the Policy object to instantiate new Permission objects.

Parameters:

`name` - the name of the ProviderPermission or "*".

`actions` - the requested actions.

org.dvb.spi

ProviderRegistry

Declaration

```
public class ProviderRegistry
```

```
java.lang.Object
  |
  +--org.dvb.spi.ProviderRegistry
```

Description

Registry of providers.

Since:

MHP 1.1.3.

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

Constructors

ProviderRegistry()

```
protected ProviderRegistry()
```

This constructor is provided for use by implementations and by other specifications that extend this class. It is not to be used by normal applications.

Methods

getInstalledProviders()

```
public java.lang.String[] getInstalledProviders()
```

Return the names of all installed providers. These are the names returned by the getName methods on those Providers. Provider names shall be encoded as defined for permission request file in the main body of the present document. For example "0x0000000B.EMV_PK11.VISA_REVOLVER".

Returns:

the names of all installed providers.

See Also:

`Provider.getName()`

getInstance()

```
public static org.dvb.spi.ProviderRegistry getInstance()
```

Return the singleton provider registry as seen by the calling application.

Returns:

the provider registry.

getProviderVersion(String)

```
public java.lang.String getProviderVersion(java.lang.String provider)
```

Return the version of an installed provider.

Parameters:

`provider` - the name of a provider as returned by the method getInstalledProviders.

Returns:

the version of the specified provider.

Throws:

`java.lang.IllegalArgumentException` - if the provider name is not one of those installed, i.e. is not one returned from a call to getInstalledProviders.

See Also:

`Provider.getVersion()`

registerSystemBound(SystemBoundProvider)

```
public void registerSystemBound(org.dvb.spi.SystemBoundProvider p)
throws ProviderFailedInstallationException
```

Registers a provider. Note that providers might be installed "automatically" by the terminal, e.g. due to signalling.

Parameters:

`p` - the provider to register.

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

Throws:

`java.lang.IllegalArgumentException` - if Provider does not expoort a valid set of services as determined by Provider.getServiceProviderInterfaces().

`ProviderFailedInstallationException` - if the organisation_id in the name of the provider does not match the organisation_id in a certificate which can authenticate the provider class.

`java.lang.SecurityException` - if the caller, for all of the SPIs implemented by the provider, does not have a ProviderPermission whose name is the fully qualified name of the class returned by Provider getService-ProviderInterfaces and whose action is "system".

See Also:

`Provider.getServiceProviderInterfaces()`

registerXletBound(XletBoundProvider)

```
public void registerXletBound(org.dvb.spi.XletBoundProvider p)
throws ProviderFailedInstallationException
```

Registers a provider. Note that providers might be installed "automatically" by the terminal, e.g. due to signalling.

Parameters:

`p` - the provider to register.

Throws:

`java.lang.IllegalArgumentException` - if Provider does not export a valid set of services as determined by Provider.getServiceProviderInterfaces(), or if the provider does not have have a non-null Xlet context.

`ProviderFailedInstallationException` - if the organisation_id in the name of the provider does not match the organisation_id in a certificate which can authenticate the provider class.

`java.lang.SecurityException` - if the caller, for all of the SPIs implemented by the provider, does not have a ProviderPermission whose name is the fully qualified name of the class returned by Provider getService-ProviderInterfaces and whose action is "xlet".

See Also:

`Provider.getServiceProviderInterfaces()`, `XletBoundProvider.getBoundXletContext()`, `XletBoundProvider.getBoundPBPXletContext()`

unregister(Provider)

```
public void unregister(org.dvb.spi.Provider p)
```

Unregister a provider. Xlets that "manually" register a provider using one of the register methods of this class shall unregister that provider before returning from a successful destroyXlet call.

Parameters:

`p` - the provider to unregister.

See Also:

`javax.tv.xlet.Xlet.destroyXlet(boolean)`

org.dvb.spi

SystemBoundProvider

Declaration

```
public interface SystemBoundProvider extends Provider
```


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

All Superinterfaces:

`Provider`

All Known Subinterfaces:

`org.dvb.spi.ict.InteractionChannelTransportProvider, org.dvb.spi.selection.SelectionProvider, org.dvb.spi.si.simple.SimpleSIProvider`

Description

Interface for providers that are registered from one Xlet, but provide system-wide services. Such providers shall be used by the system in such a way that there is no sharing of mutable object instances between the xlet that installs the provider, and any xlet that uses the services.

Since:

MHP 1.1.3.

See Also:

`XletBoundProvider`

org.dvb.spi

XletBoundProvider

Declaration

```
public interface XletBoundProvider extends Provider
```

All Superinterfaces:

`Provider`

All Known Subinterfaces:

`org.dvb.security.provider.CryptographicServiceProviderProvider, org.dvb.spi.si.full.SIManagerProvider`

Description

Interface for all Xlet-bound providers. An instance of an Xlet-bound provider provides its services for exactly one xlet: The xlet with which that provider is registered. The classes of the provider are in the classloader hierarchy of the xlet itself. This leads to a very simple mechanism, because there are no issues with instance sharing between the provider and the xlet that uses the provider. However, it has the disadvantage that every xlet that needs the services provided must carry its own copy of the provider.

Since:

MHP 1.1.3.

See Also:

`SystemBoundProvider`

Methods

getBoundPBPXletContext()

```
public javax.microedition.xlet.XletContext getBoundPBPXletContext()
```

A valid provider shall return a non-null value from at least one of the get...XletContext methods.

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

Returns:

the xlet context of the xlet to which the provider is bound.

getBoundXletContext()

```
public javax.tv.xlet.XletContext getBoundXletContext()
```

A valid provider shall return a non-null value from at least one of the get...XletContext methods.

Returns:

the xlet context of the xlet to which the provider is bound.

Package org.dvb.spi.selection

Description

This package defines an SPI for selection of services and service components. The providers defined in this package allow the presentation of services whose location is not announced using standard signalling. One a SelectionProvider has been installed, the standard GEM APIs, such as the JavaTV service selection API, can be used to present those services.

Examples of API usage

Switched Digital

The following table shows the sequence of API calls involved in a service selection operation, both API calls between an application and the implementation and between the implementation and a SelectionProvider.

The table covers both the case when the service location is known and when the location is unknown. The first column indicates if the step applies when the service location is unknown. The second column indicates whether the step applies when the service location is known. Most steps either apply in both cases at the same point in the sequence or only apply when the service location is unknown. Two steps apply in both cases but at different points in the sequence.

Unknown Service Location	Known Service Location	Step	API call by application	API call to/from provider
Y	Y	Provider is installed	`ProviderRegistry.registerSystemBound`	`Provider.providerRegistered`
Y	Y	Provider is initialised	None	`SelectionProvider.init`
Y	Y	Implementation asks provider for supported services	None	`SelectionProvider.getServiceList()`
Y	Y	Each supported service is merged into the service list of the receiver	None	Methods on each ServiceReference returned from `SelectionProvider.getServiceList()`
N	Y	Extract tuning information for service	None	For GEM, `org.dvb.locator.FrequencyLocator.getDeliverySystemDescriptor()`
Some time later				
Y	Y	Application obtains service object corresponding to locator of service	SIManager.get-Service(Locator-Factory.create-Locator("dvb://'movie-channel-1.broadcaster-b.com'")	None

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

Unknown Service Location	Known Service Location	Step	API call by application	API call to/from provider
Y	Y	Application asks for service to be presented	Service-Context.select(Service)	None
Y	N	Create session for the presentation of the requested service.	None	1) new ServiceReference(..) 2) `SelectionProvider.newSession`
Y	N	Implementation asks provider to present channel.	None	`SelectionSession.select`
Y	N	Provider asks head-end for channel	None	Existing java.net APIs
Y	N	Head-end returns reference to channel location	None	Existing java.net APIs
Y	N	Provider passes on reference to channel location to implementation	None	1a) For GEM, new Frequency-Locator(...) 1b) For OCAP, new OCAPLocator(int frequency, int programNumber,int modulationFormat, ...) 2) Return from call to `SelectionSession.select()`
Y	N	Extract tuning information for service	None	For GEM, `org.dvb.locator.FrequencyLocator.getDeliverySystemDescriptor()`
Y	Y	Implementation tunes to appropriate frequency etc	None	None
N	Y	Create session for the presentation of the requested service.	None	1) new ServiceReference(..) 2) `SelectionProvider.newSession`
Y	N	Inform provider that implementation is ready to receive content	None	`SelectionSession.selectionReady`
Y	N	Provider tells server to send content	Provider sends message to switched digital server	Existing java.net APIs
Y	Y	Content arrives	1) new Normal-ContentEvent 2) ServiceContext-Listener.receive-ServiceContext-Event(..)	None
Some time later				
Y	Y	Content stops being presented (change to new channel, ...)	Either none or Service-Context.select() with a different service.	`SelectionSession.destroy`
Y	Y	Content may stop being delivered	Provider may tell head-end that this content is no longer needed if the head-end is counting the number of users of the content	Existing java.net APIs

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

Video on Demand using Non-Standard RTSP

The table below shows the use of a org.dvb.spi.selection.SelectionProvider with RTSP when content is presented using the GEM service selection API. The rows colour coded in grey are specific to content being presented using the the service selection API. Equivalents exist if JMF is used to present content using a provider.

Step	API call by application	API call to/from provider	Notes
Provider is installed	`ProviderRegistry.registerSystemBound`	`Provider.providerRegistered`	
Provider is initialised	None	`SelectionProvider.init`	
Provider registers locator schemes it can select	None	Implementation calls `SelectionProvider.getSupportedLocatorSchemes()` which returns new org.dvb.spi.selection.Locator-Scheme("rtsp", true)	
Some time later			
Application decides which content is to be played back	new RTSPLocator(...)	None	
Application asks for content to be presented	SIManager.getService(Locator) ServiceContext.select(Service)	None	
Implementation decides whether to use built-in RTSP implementation or provider.	None	`LocatorScheme.getProviderFirst()`	
Create session for the presentation of the requested content.	None	New ServiceReference(..) `SelectionProvider.newSession`	
Provider negotiates RTSP session with server.	None	Provider uses existing java.net APIs to issue SETUP command including GEM terminal IP address and UDP port chosen by the provider to receive the MPEG stream from the RTSP server	
Provider tells implementation which port will be used for incoming content	None	Provider returns new `org.dvb.locator.ip.UnicastLocator` from call to `SelectionSession.select()`	
Implementation listens on port identified by the UDPLocator returned from the provider and sets up MPEG demux / decoder chain as appropriate	None	None	
Server sends content when GEM terminal is ready to receive it.	None	Implementation calls `SelectionSession.selectionReady()` Provider uses existing java.net APIs to send PLAY command to server	
Content arrives	1) new NormalContentEvent 2) ServiceContextListener.receive-ServiceContextEvent(..)	None	
Some time later			
Application wants to pause playback	Player.setRate(0)	Implementation calls `SelectionSession.setRate(float)` Provider uses existing java.net APIs to send RTSP pause command to server.	
Some time later			
Application wants to resume playback	Player.setRate(1.0)	Implementation calls `SelectionSession.setRate(float)` Provider uses existing java.net APIs to send RTSP PLAY command to server.	
Some time later			
Application wants playback to jump to new location	Player.setMediaTime()	Implementation calls `SelectionSession.setPosition(long)`	

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 769 of 861

Step	API call by application	API call to/from provider	Notes
		Provider uses existing java.net APIs to send PLAY command specifying new starting offset.	
Some time later			
Application wants playback to switch into fast forwards	Player.setRate()	Implementation calls `SelectionSession.setRate(float)` Provider uses existing java.net APIs to send PLAY command with desired playscale	
Some time later			
Content stops being presented (change to new channel, ...)	Either none or ServiceContext.select() with a different service.	`SelectionSession.destroy()`	
Provider sends TEARDOWN command to server and closes session	None	Existing java.net APIs	

Class Summary	
Interfaces	
`SelectionProvider`	A provider of service selection.
`SelectionProviderContext`	Platform implementations that wish to receive notifications from a `SelectionProvider` register an instance that implements this interface with the `SelectionProvider`.
`SelectionSession`	A session for presentation of one or more services over a period of time by a SelectionProvider.
`ServiceDescription`	Represents the description of a service.
Classes	
`KnownServiceReference`	A reference to a service that is reached with the mediation of a SelectionProvider.
`LocatorScheme`	This class is used to represent a locator scheme that is supported by a SelectionProvider.
`ServiceReference`	A reference to a service that is reached with the mediation of a SelectionProvider.

org.dvb.spi.selection

KnownServiceReference

Declaration

```
public class KnownServiceReference extends ServiceReference

java.lang.Object
  |
  +--org.dvb.spi.selection.ServiceReference
        |
        +--org.dvb.spi.selection.KnownServiceReference
```

Description

A reference to a service that is reached with the mediation of a SelectionProvider. This class represents services whose location is already known to the provider without needing to ask a server or head-end.

Since:

MHP 1.1.3.

See Also:

`SelectionProvider`

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

Constructors

KnownServiceReference(String, String, Locator)

```
public KnownServiceReference(java.lang.String transportIndependent,
java.lang.String transportDependent, javax.tv.locator.Locator actualLocation)
```

Create a ServiceReference for the given service identified by the combination of a transport independent and transport dependent identifiers.

Parameters:

`transportIndependent` - The transport-independent locator of the service.

`transportDependent` - The transport-dependent locator of the service.

`actualLocation` - The actual location of the service.

Methods

getActualLocation()

```
public javax.tv.locator.Locator getActualLocation()
```

Return the actualLocation as provided when this ServiceReference was constructed.

Returns:

a Locator.

getLocator()

```
public javax.tv.locator.Locator getLocator()
```

Gives the transport-dependent locator that will be used to represent this service through the Java TV APIs, both in this xlet and in others.

Overrides:

`getLocator` in class `ServiceReference`

getServiceIdentifier()

```
public java.lang.String getServiceIdentifier()
```

Description copied from class:

```
org.dvb.spi.selection.ServiceReference
```

Return the transport independent locator of the service.

Overrides:

`getServiceIdentifier` in class `ServiceReference`

Returns:

the serviceIdentifier string passed into the constructor.

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

org.dvb.spi.selection

LocatorScheme

Declaration

```
public class LocatorScheme

java.lang.Object
  |
  +--org.dvb.spi.selection.LocatorScheme
```

Description

This class is used to represent a locator scheme that is supported by a SelectionProvider.

Since:

MHP 1.1.3.

Constructors

LocatorScheme(String, boolean)

```
public LocatorScheme(java.lang.String scheme, boolean providerFirst)
```

Construct a LocatorScheme.

Parameters:

`scheme` - the name of the locator scheme which is handled, not including the ":" character.

`providerFirst` - true if this provider shall handle the scheme before any handler in the platform implementation, false if this provider shall handle the scheme after any handler in the platform implementation and only for locators not handled by the platform implementation.

Methods

getProviderFirst()

```
public boolean getProviderFirst()
```

Returns whether the provider is to handle the scheme before any platform handler for the scheme.

Returns:

true if this provider shall handle the scheme before any handler in the platform implementation, false if this provider shall handle the scheme after any handler in the platform implementation and only for locators not handled by the platform implementation.

getScheme()

```
public java.lang.String getScheme()
```

Returns the scheme.

Returns:

scheme the name of the locator scheme which is handled, not including the ":" character.

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

org.dvb.spi.selection

SelectionProvider

Declaration

```
public interface SelectionProvider extends org.dvb.spi.SystemBoundProvider
```

All Superinterfaces:

```
org.dvb.spi.Provider, org.dvb.spi.SystemBoundProvider
```

Description

A provider of service selection. For example, a SelectionProvider can be installed in an always-running service-unbound xlet to provide the ability to reach services that do not have standarized SI signalling in an IPTV network.

The process for determining which Provider (if any) shall be used to present a transport independent service is defined by the following steps in the order given.

1) Consult the service list. If the service appears in the service list then make an selection between the available sources of that Service as defined by ServiceContext.select(Service).

2) If there one Provider registered to support the locator scheme of the locator for the Service, ask that Provider to present that service. The service selection operation shall fail if the provider cannot present it.

3) If more than one Provider has registed as supporting the locator scheme of the locator for the Service, offer the Service to each Provider in turn until either one of them does not fail or no more Providers are available in which case the service selection operation shall fail.

NOTE: If the performance penalty of polling multiple Providers for the same locator scheme is undesirable, service providers / operators using multiple Providers should ensure they use different Locator schemes.

Providers should select the transport dependent locators returned such that there is no collision the transport dependent locators returned by other Providers that may offer the same transport independent service. Where a GEM terminal is managed by a service provider / operator, the service provider / operator should enforce this if they permit more than one Provider to be installed.

Where there is a collision and a transport dependent locator can be accessed by more than one Provider, the selection of which to use is implementation dependent.

A SelectionProvider may inform the implementation of the actual location of a service by two mechanisms.

- For services which are already present in the network, the actual location can be passed in ServiceReference objects. These can be provided by the getServiceList method when a provider is first installed and later by the SelectionProviderContext. For these services, the implementation shall use this actual location to present the service without calling the select method. A session shall be created using the ServiceReference used to present the service in order that the provider can be notified when the service is no longer used. The session instance should be created once the platform has started finding the service (e.g. during tuning or after sending an IGMP join request) in order to avoid delaying the presentation of the service.

- For services whose actual location is not passed in a ServiceReference (e.g. services not already present in the network), the implementation shall use the select method to obtain the actual location.

The mechanism by which the actual location of a service is determined may change dynamically, e.g. as the set of services available in the network changes.

Since:

MHP 1.1.3.

See Also:

```
org.dvb.spi.Provider.getServiceProviderInterfaces()
```


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

Methods

getServiceDescriptions(ServiceReference[])

```
public org.dvb.spi.selection.ServiceDescription[]
getServiceDescriptions(org.dvb.spi.selection.ServiceReference[] services)
```

Called by the terminal to request service description information from an SI source. The output shall be returned in an array of the same size and order as the input. If information is not available on any service listed in the input then the corresponding entry in the results array shall be null.

Parameters:

`services` - References identifying the services whose descriptions are requested.

Returns:

an array of service descriptions.

getServiceList()

```
public org.dvb.spi.selection.ServiceReference[] getServiceList()
```

Give a list of the services provided by this provider. The services returned from this method shall be merged into the platform service list as returned by javax.tv.service.SIManager.filterServices (null). Where the transport independent identification of a service is equal to one already in the service list then that transport independent service shall acquire an additional transport dependent service. Where the transport independent identification of a service is not equal to one already in the service list, a new service shall be added to the service list and SIChangeEvents generated to appropriate listeners.

The list of services returned shall include both those where the actual location is returned in the ServiceReference and those where the actual location is to be returned from a later call to the select method.

Returns:

a list of ServiceReference instances.

See Also:

`SelectionProviderContext.serviceListChanged(ServiceReference[])`

getSupportedLocatorSchemes()

```
public org.dvb.spi.selection.LocatorScheme[] getSupportedLocatorSchemes()
```

Returns the list of locator schemes handled by this provider. This list should not change over time; it is expected that platforms will usually call this method exactly once, after installation of a provider. There is no method to unregister a scheme other than unregistering the provider.

init(SelectionProviderContext)

```
public void init(org.dvb.spi.selection.SelectionProviderContext c)
```

Called by the platform to register its handler for events originating in the provider. This method shall be called after providerRegistered and before any other methods on the provider are called. Only one handler can be registered at any one time.

Parameters:

`c` - the handler.

newSession(ServiceReference)

```
public org.dvb.spi.selection.SelectionSession newSession(org.dvb.spi.selection.ServiceReference
service)
```

Called by the platform to create a session to manage the presentation of a service. A new session shall be created for each service to be presented by a provider.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 774 of 861

Parameters:

`service` - the service whose presentation is managed through this session.

Returns:

a session.

org.dvb.spi.selection SelectionProviderContext

Declaration

```
public interface SelectionProviderContext
```

Description

Platform implementations that wish to receive notifications from a `SelectionProvider` register an instance that implements this interface with the `SelectionProvider`.

Since:

MHP 1.1.3.

Methods

serviceDescriptionAvailable(ServiceReference[])

```
public void serviceDescriptionAvailable(org.dvb.spi.selection.ServiceReference[] serviceReferences)
```

Called by a source when service description information is available to offer that information to the platform implementation. The new list replaces any previous list.

Parameters:

`serviceReferences` - The ServiceReference instances for which descriptions are available. This array must not change after this method is invoked.

serviceListChanged(ServiceReference[])

```
public void serviceListChanged(org.dvb.spi.selection.ServiceReference[] serviceReferences)
```

Called by a source when the list of services changes. The new list replaces any previous list.

The services passed into this method shall be compared with the service list returned from the call to getServiceList when this provider was first registered. Where services are added, these shall be merged into the platform service list as defined in the description of the getServiceList method. Where services are removed, if the transport independent service is left with no transport dependent services then it shall be removed from the platform service list. In all cases where the platform service list changes, SIChangeEvents shall be generated to appropriate listeners.

Parameters:

`serviceReferences` - The ServiceReference instances for the available services. This array must not change after this method is invoked.

updateService(ServiceReference)

```
public void updateService(org.dvb.spi.selection.ServiceReference service)
```

Called by a source to update the details of a service it supports. This includes changing the service between one whose location is already known and one whose location must be found when it is selected. It also includes changing the location of a service whose location is already known. ServiceReferences shall be matched using the transport independent and transport dependent names.

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

Parameters:

`service` - the new service reference.

Throws:

`java.lang.IllegalArgumentException` - if a service with the same service identifier and transport independent locator has not been previously returned by this source to the implementation.

org.dvb.spi.selection

SelectionSession

Declaration

```
public interface SelectionSession
```

Description

A session for presentation of one or more services over a period of time by a SelectionProvider. The first operation on a new session will always be selection of a service in a service context, and all subsequent operations will pertain to the same service context.

Since:

MHP 1.1.3.

Methods

destroy()

```
public void destroy()
```

Called by the platform when the service bound to this session is no longer being used by the implementation. The selection provider should do any needed clean-up here, e.g. informing a server that the service is not needed any more. Once destroy is called, the terminal implementation discards the SelectionSession.

select()

```
public org.davic.net.Locator select()
```

Sets up delivery of a stream representing the service, and delivers a locator for reception of that stream. For example, either a unicast locator or a frequency locator might be returned, under the apprlpriate circumstances.

Returns:

a locator for where the stream can be found.

selectionReady()

```
public void selectionReady()
```

Called when the implementation is ready to receive content on the locator returned by the select method. When using unicast protocols where the content must only be sent when the GEM terminal is ready, it is now safe to request the content be sent.

A SelectionSession might be destroyed after a select, but before this method is called. In this case, this method may not be called. Applications should therefore not expect this method to be called after destroy() is called on this session.

See Also:

`destroy()`

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

setPosition(long)

```
public long setPosition(long position)
```

Set the position within the media. If this session does not support trick modes, this method returns -1 and has no effect.

Parameters:

`position` - The position within the program, in milliseconds.

Returns:

The new position, or -1 if position setting isn't supported.

setRate(float)

```
public float setRate(float newRate)
```

Sets the speed of playback. If this session does not support trick modes, this method returns 1 and has no effect. Calling this method with the value 1.0f shall always succeed, and shall result in a return value of 1.0f.

Parameters:

`newRate` - New playback rate. Implementations shall make a best effort to approximate this rate.

Returns:

The new rate, if known. If not known, the value java.lang.Float.NEGATIVE_INFINITY is returned.

org.dvb.spi.selection

ServiceDescription

Declaration

```
public interface ServiceDescription
```

Description

Represents the description of a service.

Since:

MHP 1.1.3.

Methods

getDeliverySystemType()

```
public javax.tv.service.navigation.DeliverySystemType getDeliverySystemType()
```

Return the type of the delivery system.

Returns:

the type of the delivery system by which this service is delivered.

Throws:

`java.lang.IllegalArgumentException` - if the returned ServiceType is not from a class loaded by the system classloader.

getLongName(String)

```
public org.dvb.spi.util.MultilingualString getLongName(java.lang.String perferredLanguage)
```

Return the long name of this service.

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

Returns:

the name.

getServiceType()

```
public javax.tv.service.ServiceType getServiceType()
```

Return the type of this service. The service type returned shall be from a class loaded by the system classloader.

Returns:

the service type of this service.

Throws:

`java.lang.IllegalArgumentException` - if the returned ServiceType is not from a class loaded by the system classloader.

org.dvb.spi.selection

ServiceReference

Declaration

```
public class ServiceReference
```

```
java.lang.Object
  |
  +--org.dvb.spi.selection.ServiceReference
```

Direct Known Subclasses:

`KnownServiceReference`

Description

A reference to a service that is reached with the mediation of a SelectionProvider.

Since:

MHP 1.1.3.

See Also:

`SelectionProvider`

Constructors

ServiceReference(String, String)

```
public ServiceReference(java.lang.String transportIndependent, java.lang.String transportDependent)
```

Create a ServiceReference for a service. The two strings shall both be the external form of Locators, for example a transport independent "dvb:" locator string and provider specific locator string.

Parameters:

`transportIndependent` - The transport-independent locator of the service.

`transportDependent` - The transport-dependent locator of the service.

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

Methods

getLocator()

```
public javax.tv.locator.Locator getLocator()
```

Gives the transport-dependent locator of the service

Returns:

the trasnport dependent string passed into the constructor.

getServiceIdentifier()

```
public java.lang.String getServiceIdentifier()
```

Return the transport independent locator of the service

Returns:

the transportIndependent string passed into the constructor.

Package org.dvb.spi.si.full

Description

This package defines an interface for providing full service information, as accessed by javax.tv.service.SIManager. This package defines an xlet-bound provider. Since it's an xlet-bound manager, every xlet that needs the services of this kind of provider needs to carry its own implementation of the provider. It is envisaged that the provider defined in the package `org.dvb.si.simple` will be used more frequently.

Class Summary	
Interfaces	
SIManagerProvider	This xlet-bound provider allows one to take complete control of the view of SI exposed to the xlet to which it is bound.

org.dvb.spi.si.full SIManagerProvider

Declaration

```
public interface SIManagerProvider extends org.dvb.spi.XletBoundProvider
```

All Superinterfaces:

```
org.dvb.spi.Provider, org.dvb.spi.XletBoundProvider
```

Description

This xlet-bound provider allows one to take complete control of the view of SI exposed to the xlet to which it is bound. The author of the provider is able to directly implement Java TV's SIManager interface, and provide that instance to the implementation. The xlet to which this provider is bound will get this instance of SIManager when it calls the Java TV method to retrieve an SIManager.

See Also:

```
javax.tv.service.SIManager.createInstance()
```


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

Methods

createSIManager(SIManager)

```
public javax.tv.service.SIManager createSIManager(javax.tv.service.SIManager parent)
```

Gets an SI manager. This will typically be given directly to the xlet (when it calls SIManager.createInstance()). This method will be called each time the xlet calls SIManager.createInstance().

Parameters:

`parent` - The "system" SIManager that would have been given to the xlet, if a provider had not been installed. The provider's SIManager may consult its parent, e.g. for terrestrial services in a hybrid terminal.

Package org.dvb.spi.si.simple

Description

This package defines interfaces for providing service information, as accessed by SIManager. This package defines a system-bound provider. As a system-bound provider, only one provider needs to be registered with the GEM terminal, and this one provider services multiple xlets. It provides the minimal SI believed to be adequate to support service-bound xlets that rely on the standard Java TV SI access mechanisms.

Class Summary	
Interfaces	
`ProgramDescription`	Encapsulates the description of a program.
`ScheduleDescription`	Encapsulates the description of a schedule for a service (sometimes colloquially known as a "TV channel").
`SimpleSIProvider`	This interface describes a source of information about the schedules and program events in a broadcast channel.
`SimpleSIProviderListener`	An instance of this class is given to a SimpleSIProvider when registered so that the provider can notify the platform of events that happen.
Classes	
`ProgramReference`	This class represents a reference to a program within a service that is accessed using a SelectionProvider.
`ProgramReferenceWithTimes`	Wraps a program reference with its start and end times.

org.dvb.spi.si.simple ProgramDescription

Declaration

```
public interface ProgramDescription
```

Description

Encapsulates the description of a program. One instance of ProgramDescription represents a static snapshot of the information about a single program.

The program descriptions may be relatively large. Implementations with limited memory resources should be careful not to ask for many descriptions at one time and not to keep references to descriptions longer than needed.

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

Methods

getDescription(String)

```
public org.dvb.spi.util.MultilingualString getDescription(java.lang.String preferredLanguage)
```

Return the description of the program.

Parameters:

`preferredLanguage` - the preferred language for the description.

Returns:

the description.

getEndTime()

```
public java.util.Date getEndTime()
```

Return the end time for the program.

Returns:

the end time.

getName(String)

```
public org.dvb.spi.util.MultilingualString getName(java.lang.String preferredLanguage)
```

Return the name of the program.

Parameters:

`preferredLanguage` - the preferred language for the name.

Returns:

the name.

getStartTime()

```
public java.util.Date getStartTime()
```

Return the start time for the program.

Returns:

the start time.

org.dvb.spi.si.simple ProgramReference

Declaration

```
public final class ProgramReference

java.lang.Object
  |
  +--org.dvb.spi.si.simple.ProgramReference
```

Description

This class represents a reference to a program within a service that is accessed using a SelectionProvider.

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

See Also:

org.dvb.spi.selection.SelectionProvider, SimpleSIProvider.getProgramReferences(ServiceReference), ProgramDescription

Constructors

ProgramReference(ServiceReference, String)

```
public ProgramReference(org.dvb.spi.selection.ServiceReference service,
java.lang.String programIdentifier)
```

Constructs a reference to a program.

Parameters:

service - the service containing the program.

programIdentifier - an identifier for the program within the service.

Methods

getProgramIdentifier()

```
public java.lang.String getProgramIdentifier()
```

Get the identifier of the program.

Returns:

a program identifier.

getServiceReference()

```
public org.dvb.spi.selection.ServiceReference getServiceReference()
```

Get the service containing the program as passed to the constructor.

Returns:

a service.

org.dvb.spi.si.simple

ProgramReferenceWithTimes

Declaration

```
public final class ProgramReferenceWithTimes
```

```
java.lang.Object
  |
  +--org.dvb.spi.si.simple.ProgramReferenceWithTimes
```

Description

Wraps a program reference with its start and end times.

Constructors

ProgramReferenceWithTimes(ProgramReference, Date, Date)

```
public ProgramReferenceWithTimes(org.dvb.spi.si.simple.ProgramReference r, java.util.Date start,
java.util.Date end)
```

Construct the program reference with times.

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

Parameters:

`r` - the program reference.

`start` - the (inclusive) start time of the program.

`end` - the (exclusive) end time of the program.

Methods

getEndTime()

```
public java.util.Date getEndTime()
```

Return the end time of the program as passed in to the constructor.

Returns:

a time.

getProgramReference()

```
public org.dvb.spi.si.simple.ProgramReference getProgramReference()
```

Return the program reference as passed in to the constructor.

Returns:

a program reference.

getStartTime()

```
public java.util.Date getStartTime()
```

Return the start time of the program as passed in to the constructor.

Returns:

a time.

org.dvb.spi.si.simple

ScheduleDescription

Declaration

```
public interface ScheduleDescription
```

Description

Encapsulates the description of a schedule for a service (sometimes colloquially known as a "TV channel"). One instance of ScheduleDescription represents a static snapshot of one time-bounded part of the schedule information. The schedule shall be sorted from earliest to latest within the schedule.

Methods

getPrograms()

```
public java.util.List getPrograms()
```

Return identifiers for the programs in the service, along with the start and end times.

Returns:

a list of ProgramReferenceWithTime instances to the programs in the service sorted in time order from first to last. This list instance must not change after this method is invoked.

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

org.dvb.spi.si.simple

SimpleSIProvider

Declaration

```
public interface SimpleSIProvider extends org.dvb.spi.SystemBoundProvider
```

All Superinterfaces:

```
org.dvb.spi.Provider, org.dvb.spi.SystemBoundProvider
```

Description

This interface describes a source of information about the schedules and program events in a broadcast channel.

This class and related classes and interfaces are intended to be used as follows:

- When new program data is available, the SI source calls the method on SimpleSIProviderContext to tell the platform which entities have data available about them.
- The platform can, at any time, ask the provider for information on certain entities. This may be in response to the provider telling the platform that new information is available. It may be in response to an application asking for information about an entity whose description is not available in the terminal.

In this API, collections of data are represented with random-access java.util.List type.

This class involves several methods that pass an instance of List to the terminal. Once a list instance has been give to the terminal, it must not be changed. The results of changing a list that has been given to the terminal via this API are undefined, and may include termination of the enclosing xlet.

See Also:

`org.dvb.spi.SystemBoundProvider`, `org.dvb.spi.selection.SelectionProvider`, `java.util.List`, `java.util.ArrayList`

Methods

getProgramDescriptions(ProgramReference[], String[])

```
public org.dvb.spi.si.simple.ProgramDescription[]
getProgramDescriptions(org.dvb.spi.si.simple.ProgramReference[] programReferences,
java.lang.String[] preferredLanguages)
```

Called by a terminal to request program description information from a provider. The output shall be returned in an array of the same size and order as the input. If the schedule is not available on any service listed in the input then the corresponding entry in the results array shall be null.

Parameters:

`programReferences` - references identifying the programs whose schedules are requested.

`preferredLanguages` - The preferred language(s) for program names and descriptions. This is a hint of the language(s) likely to be requested of the descriptions themselves.

Returns:

an array of program descriptions.

getProgramReferences(ServiceReference)

```
public org.dvb.spi.si.simple.ProgramReference[]
getProgramReferences(org.dvb.spi.selection.ServiceReference r)
```

Called by the terminal to request program information from a provider for a given service.

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

Parameters:

r - the service for which program information is requested.

Returns:

references for all of the known programs for the given service.

getScheduleDescriptions(ServiceReference[], Date, Date)

```
public org.dvb.spi.si.simple.ScheduleDescription[]
getScheduleDescriptions(org.dvb.spi.selection.ServiceReference[] services, java.util.Date startDate,
java.util.Date endDate)
```

Called by the terminal to request schedule information from a provider. The output shall be returned in an array of the same size and order as the input. If the schedule is not available on any service listed in the input then the corresponding entry in the results array shall be null.

Parameters:

services - references identifying the services whose schedules are requested.

startDate - the starting date and time from which to return schedule descriptions (inclusive).

endDate - the finishing date and time up to which schedule descriptons should be returned (exclusive).

Returns:

an array of schedule descriptions.

registerListener(SimpleSIProviderListener)

```
public void registerListener(org.dvb.spi.si.simple.SimpleSIProviderListener l)
```

Called by the platform in order to obtain events from this provider.

Parameters:

l - the listener to register.

unregisterListener(SimpleSIProviderListener)

```
public void unregisterListener(org.dvb.spi.si.simple.SimpleSIProviderListener l)
```

Called by the platform when it no longer wishes to receive events from this provider.

Parameters:

l - the listener to remove.

org.dvb.spi.si.simple SimpleSIProviderListener

Declaration

```
public interface SimpleSIProviderListener
```

Description

An instance of this class is given to a SimpleSIProvider when registered so that the provider can notify the platform of events that happen.

See Also:

SimpleSIProvider.registerListener(SimpleSIProviderListener)

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

Methods

scheduleDescriptionAvailable(List)

```
public void scheduleDescriptionAvailable(java.util.List serviceReferences)
```

Called by providers when schedule information is available to offer that information to the terminal.

Parameters:

`serviceReferences` - A list of ServiceReference instances for which descriptions are available. This list instance must not change after this method is invoked.

Package org.dvb.spi.util

Description

Utility classes used by other providers.

Class Summary	
Classes	
`MultilingualString`	A string that has an associated language code.

org.dvb.spi.util MultilingualString

Declaration

```
public final class MultilingualString

java.lang.Object
  |
  +--org.dvb.spi.util.MultilingualString
```

Description

A string that has an associated language code.

Since:

MHP 1.1.3.

Constructors

MultilingualString(String, String)

```
public MultilingualString(java.lang.String s, java.lang.String languageCode)
```

Initialize a new MultilingualString.

Parameters:

`s` - The string that is in the given language.

`languageCode` - The language code encoded as per ISO 639-2 [84].

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

Methods

getLanguageCode()

```
public java.lang.String getLanguageCode()
```

Return the language code.

Returns:

The language code encoded as per ISO 639-2 [84].

getString()

```
public java.lang.String getString()
```

Return the string.

Returns:

The string that is in the given language.

toString()

```
public java.lang.String toString()
```

Return a string representation of this object.

Overrides:

`toString` in class `Object`

Returns:

a string.

Package org.dvb.spi.ict

Description

This package enables the GEM return channel application download mechanism to be used with non-standard, proprietary or market specific protocols.

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

Class Summary	
Interfaces	
AsyncResourceLoadListener	This interface should be implemented by classes within the GEM implementation that will be used as listeners for asynchronous loading of resources by interaction channel transport service providers.
InteractionChannelTransportProvider	This interface will be implemented by system providers of resource transport via the interaction channel.
ResourceTransportObject	This interface represents a reference to a resource that will be downloaded using an interaction channel transport service provider.
ResourceUpdateListener	This interface will be implemented by listeners that are part of the GEM implementation so that they are notified of updates to resources downloaded by an interaction channel tranport service provider.
Classes	
AsyncResourceLoadEvent	Base class for events related to progress of loading an event by an interaction channel transport service provider.
LoadFailedEvent	Event indicating that the load of a resource failed.
LoadFinishedEvent	Event indicating that the load of a resource was completed successfully.
ResourceObjectUpdateEvent	Event object used to notify update listeners that a network resource has changed (i.e. a new version is available).
Exceptions	
ResourceLoadException	This exception is thrown when an attempt to load a resource by an interaction channel transport service provider fails.
ResourceNotFoundException	This exception is thrown when an attempt is made to load a resource via interaction channel transport, and the provider determines that the resource does not exist.

org.dvb.spi.ict

AsyncResourceLoadEvent

Declaration

```
public class AsyncResourceLoadEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.spi.ict.AsyncResourceLoadEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Direct Known Subclasses:

```
LoadFailedEvent, LoadFinishedEvent
```

Description

Base class for events related to progress of loading an event by an interaction channel transport service provider.

Constructors

AsyncResourceLoadEvent(ResourceTransportObject, boolean)

```
protected AsyncResourceLoadEvent(org.dvb.spi.ict.ResourceTransportObject source, boolean loaded)
```

Create a new AsyncResourceLoadEvent object.

Parameters:

`source` - the ResourceTransportObject that is the source of this event.

`loaded` - flag indicating whether the resource is loaded as of the receipt of this event.

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

Methods

isLoaded()

```
public boolean isLoaded()
```

Return a flag indicating whether the resource is loaded.

Returns:

boolean.

org.dvb.spi.ict AsyncResourceLoadListener

Declaration

```
public interface AsyncResourceLoadListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

This interface should be implemented by classes within the GEM implementation that will be used as listeners for asynchronous loading of resources by interaction channel transport service providers.

Note that in the case where a DVB-J application calls asynchronousLoad() on a DSMCCObject that refers to an interaction channel file system file, the GEM implementation will be respsonsible for mapping receiveEvent method calls from this method to calls on the corresponding AsynchronousLoadingEventListener (i.e. applications will not be able to reference the service provider classes directly).

Methods

receiveEvent(AsyncResourceLoadEvent)

```
public void receiveEvent(org.dvb.spi.ict.AsyncResourceLoadEvent e)
```

Notify this listener of an update in the progress of loading a network resource. The specified event object contains detailed information about the update.

Parameters:

`e` - the event object containing details of the update.

org.dvb.spi.ict InteractionChannelTransportProvider

Declaration

```
public interface InteractionChannelTransportProvider extends org.dvb.spi.SystemBoundProvider
```

All Superinterfaces:

```
org.dvb.spi.Provider, org.dvb.spi.SystemBoundProvider
```


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

Description

This interface will be implemented by system providers of resource transport via the interaction channel. It will be used by the GEM implementation when accessing resources identified by references (URIs) encountered within the interaction channel protocol descriptor. In addition, this provider may be accessed indirectly from DVB-J applications via the java.io.File and org.dvb.dsmcc.DSMCCObject classes.

Methods

createResourceTransportObject(String)

```
public org.dvb.spi.ict.ResourceTransportObject createResourceTransportObject(java.lang.String uri)
```

Create new resource access object in preparation for downloading and using the contents of a network resource via the interaction channel.

Parameters:

`uri` - URI identifying the resource to be accessed, the URI scheme of which is guaranteed to be one of those returned from getSupportedURISchemes.

Returns:

ResourceTransportObject - the object used to track download of the resource.

exists(String)

```
public boolean exists(java.lang.String uri)
```

Return a flag indicating whether the specified resource exists. The expectation is that this method will provide a more efficient indication of existence than creating a ResourceTransportObject and initiating a download, for service provider implementations that support caching of directory lookup information.

Parameters:

`uri` - URI identifying the resource whose existence to check, the URI scheme of which is guaranteed to be one of those returned from `getSupportedURISchemes()`

Returns:

boolean - Flag indicating whether the specified resource exists. Note that there is no guarantee that the resource will subsequently be loaded successfully, since the return value will be determined just from the directory information. It is possible that the resource will be found not to exist during the process of downloading it.

Throws:

`java.lang.UnsupportedOperationException` - if this provider does not support optimized existence checks (i.e. if `existsSupported()` returns `false`).

existsSupported()

```
public boolean existsSupported()
```

Return a flag indicating whether this provider supports existence checks using the exists() method on the provider implementation. This is intended as a short circuit by which the GEM implementation may avoid the overhead of creating a ResourceTransportObject instance in cases where the provider is capable of determining the existence of a resource without actually trying to load it (e.g. broadcast transport mechanisms where there may be cached directory lookup information for the resources). The goal is to optimize the performance of the logical file system search path.

Returns:

boolean - Flag indicating whether fast existence checks are supported via the `exists()` method.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 790 of 861

getSupportedURISchemes()

```
public java.lang.String[] getSupportedURISchemes()
```

Return the list of URI schemes supported by this provider. This method will be called by the GEM implementation exactly once immediately subsequent to registration of the provider. The GEM implementation will only call createResourceTransport on this provider for URI schemes that are returned by this method. In the case where multiple providers return the same URI scheme, the provider that will be used will be the one that most recently registered.

Note that if http or https are contained in the returned list they will be ignored. In other words, the service provider cannot override the GEM implementation's internal implementation of the HTTP and HTTPS protocols.

Returns:

String[] - list of supported URI schemes, each of which must consist of alphabetic characters only (no trailing ":").

prefetch(String, int)

```
public void prefetch(java.lang.String uri, int priority)
```

Provides a hint to the provider that the resource identified by the specified URI will probably be accessed in the near future, so that if available resources permit and the provider supports prefetching it can take whatever steps are appropriate to speed up the time required to load the file when the subsequent request to load the file is made via the normal call to `ResourceTransportObject.synchronousLoad()` or `ResourceTransportObject.asynchronousLoad()`. This could include anything from retrieving directory information to actually downloading the file to memory or local physical file system, if space is available.

Parameters:

`uri` - URI identifying the resource to be accessed, the URI scheme of which is guaranteed to be one of those returned from getSupportedURISchemes.

`priority` - hint to the provider about the relative priority of this resources with respect to other resources prefetch hints for this provider (higher = more important).

Throws:

`java.lang.UnsupportedOperationException` - if this provider does not support prefetching (i.e. if `prefetchSupported()` returns `false`).

prefetchSupported()

```
public boolean prefetchSupported()
```

Return a flag indicating whether this provider supports prefetch. Note that a return value of `true` does not guarantee that a call to `prefetch()` for a particular resource will have any effect. It is merely an indication of whether the provider supports prefetch hints in general.

Returns:

boolean - Flag indicating whether this provider supports prefetch hints via the `prefetch()` method.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 791 of 861

org.dvb.spi.ict
LoadFailedEvent

Declaration

```
public final class LoadFailedEvent extends AsyncResourceLoadEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.spi.ict.AsyncResourceLoadEvent
              |
              +--org.dvb.spi.ict.LoadFailedEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Event indicating that the load of a resource failed.

Constructors

LoadFailedEvent(ResourceTransportObject, Exception)

```
public LoadFailedEvent(org.dvb.spi.ict.ResourceTransportObject source, java.lang.Exception reason)
```

Construct a new event indicating failure to load a resource.

Parameters:

`source` - The ResourceTransportObject that was the source of the failure.

`reason` - Exception indicating the reason for the failure (may be null).

Methods

getReason()

```
public java.lang.Exception getReason()
```

Return the reason for the load failure.

Returns:

Exception - The exception object indicating the reason for the failure (may be null).

org.dvb.spi.ict
LoadFinishedEvent

Declaration

```
public final class LoadFinishedEvent extends AsyncResourceLoadEvent
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.spi.ict.AsyncResourceLoadEvent
              |
              +--org.dvb.spi.ict.LoadFinishedEvent
```


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

All Implemented Interfaces:

`java.io.Serializable`

Description

Event indicating that the load of a resource was completed successfully.

Constructors

LoadFinishedEvent(ResourceTransportObject)

```
public LoadFinishedEvent(org.dvb.spi.ict.ResourceTransportObject source)
```

Construct a new event indicating that the load of a resource was completed successfully.

Parameters:

`source` - The ResourceTransportObject that was the source of this event.

LoadFinishedEvent(ResourceTransportObject, boolean)

```
public LoadFinishedEvent(org.dvb.spi.ict.ResourceTransportObject source, boolean loaded)
```

Construct a new event indicating that the load of a resource was completed successfully.

Parameters:

`source` - The ResourceTransportObject that was the source of this event.

`loaded` - flag indicating whether the resource is loaded as of the receipt of this event.

org.dvb.spi.ict

ResourceLoadException

Declaration

```
public class ResourceLoadException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.spi.ict.ResourceLoadException
```

All Implemented Interfaces:

`java.io.Serializable`

Direct Known Subclasses:

`ResourceNotFoundException`

Description

This exception is thrown when an attempt to load a resource by an interaction channel transport service provider fails.

Constructors

ResourceLoadException()

```
public ResourceLoadException()
```

Create a ResourceLoadException.

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

ResourceLoadException(String)

```
public ResourceLoadException(java.lang.String message)
```

Create a ResourceLoadException.

Parameters:

`message` - describes the failure.

ResourceLoadException(String, Throwable)

```
public ResourceLoadException(java.lang.String message, java.lang.Throwable cause)
```

Create a ResourceLoadException.

Parameters:

`message` - describes the failure.

`cause` - details the cause of the failure.

ResourceLoadException(Throwable)

```
public ResourceLoadException(java.lang.Throwable cause)
```

Create a ResourceLoadException.

Parameters:

`cause` - details the cause of the failure.

org.dvb.spi.ict ResourceNotFoundException

Declaration

```
public class ResourceNotFoundException extends ResourceLoadException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--org.dvb.spi.ict.ResourceLoadException
                    |
                    +--org.dvb.spi.ict.ResourceNotFoundException
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

This exception is thrown when an attempt is made to load a resource via interaction channel transport, and the provider determines that the resource does not exist.

Constructors

ResourceNotFoundException(String)

```
public ResourceNotFoundException(java.lang.String uri)
```

Create a ResourceNotFoundException.

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

Parameters:

`uri` - the URI for the resource that was not found.

ResourceNotFoundException(String, String)

```
public ResourceNotFoundException(java.lang.String uri, java.lang.String message)
```

Create a ResourceNotFoundException.

Parameters:

`message` - describes the failure.

org.dvb.spi.ict

ResourceObjectUpdateEvent

Declaration

```
public class ResourceObjectUpdateEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.spi.ict.ResourceObjectUpdateEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

Event object used to notify update listeners that a network resource has changed (i.e. a new version is available).

Constructors

ResourceObjectUpdateEvent(ResourceTransportObject, int)

```
public ResourceObjectUpdateEvent(org.dvb.spi.ict.ResourceTransportObject source, int newVersion)
```

Create a ResourceObjectUpdateEvent indicating that a new version of the resource has been detected. Note that the new version of the resource is not automatically downloaded; that is left to the user of this object.

Parameters:

`source` - the ResourceTransportObject whose version has changed.

`newVersion` - the new version number.

Methods

getNewVersion()

```
public int getNewVersion()
```

Return the newly available resource version number (i.e. the version that would be downloaded now if the resource were loaded again).

Returns:

int

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 795 of 861

org.dvb.spi.ict

ResourceTransportObject

Declaration

```
public interface ResourceTransportObject
```

Description

This interface represents a reference to a resource that will be downloaded using an interaction channel transport service provider. It exposes a set of methods that correspond with those in org.dvb.dsmcc.DSMCCObject, so that the applications can use DSMCCObject to reference resources that will be downloaded by the GEM implementation via the service provider implementation.

Like DSMCCObject, ResourceTransportObject instances can either be in a loaded or unloaded state. When in a loaded state, the contents of the resource can be accessed either directly from memory as a byte array or else from a physical file in a local file system (could also be a RAM-based file system).

Note that unlike DSMCCObject instances, resource transport objects are only used to download resources with associated content (i.e. not for URIs that represent directories).

Fields

LOAD_TYPE_FILE

```
public static final int LOAD_TYPE_FILE
```

Resource load type that indicates the file has been loaded to a local physical file system.

LOAD_TYPE_MEMORY

```
public static final int LOAD_TYPE_MEMORY
```

Resource load type that indicates the file has been loaded in memory as a byte array.

Methods

abort()

```
public void abort()
```

This method is used to abort a load in progress. It can be used to abort either a synchronousLoad or an asynchronous-Load.

Throws:

`java.lang.IllegalStateException` - if load of the resource has not already been started.

addUpdateListener(ResourceUpdateListener)

```
public void addUpdateListener(org.dvb.spi.ict.ResourceUpdateListener listener)
```

Add a listener to receive notifications of updates to this resource. Note that update notifications will not be fired until after the object has succcessfully entered the loaded state for the first time.

Once an object has successfully entered the loaded state once, this event shall continue to be fired when changes are detected regardless of further transitions in or out of the loaded state.

Parameters:

`listener` - the listener to be added.

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

asynchronousLoad(AsyncResourceLoadListener)

```
public void asynchronousLoad(org.dvb.spi.ict.AsyncResourceLoadListener listener)
```

Load the resource asynchronously. The specified listener will be notified of the loading progress, including subsequent successful completion or failure.

Parameters:

`listener` - the listener to be notified.

exists()

```
public boolean exists()
```

Return a flag indicating whether the resource exists. Note that some providers may be able to determine resource existence without downloading the resource using cached directory lookup information, but in general this may not be the case. In cases where the provider must download the resource to determine resource existence, this call may result in a synchronous load operation being performed before returning.

Returns:

boolean.

getLength()

```
public long getLength()
```

Return the length in bytes of content of this resource. Note that some providers may be able to determine the size without downloading the resource using cached directory lookup informatino, but in general this may not be the case. In cases where the provider must download the resource to determine the length, this call may result in a synchronous load operation being performed before returning. In addition, since this information may be populated from a directory, it is not guaranteed to be accurate; the most authoritative information is specified after download from either `getResource-Bytes().length` or `getResourceFile().length()`, depending on the load type.

Returns:

long the length in bytes of the resource content.

getMimeType()

```
public java.lang.String getMimeType()
```

Return the MIME type of the resource. Note that some providers may be able to determine the MIME type without downloading the resource using cached directory lookup information, but in general this may not be the case. In cases where the provider must download the resource to determine the MIME type, this call may result in a synchronous load operation being performed before returning.

Returns:

String.

getResourceBytes()

```
public byte[] getResourceBytes()
```

Return a byte array containing the downloaded resource. This method should only be called if the resource is loaded and `getResourceLoadType()` returns `LOAD_TYPE_MEMORY`.

Returns:

byte [] the resource data.

Throws:

`java.lang.IllegalStateException` - if the resource is not loaded or is loaded to a file.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 797 of 861

getResourceFile()

```
public java.io.File getResourceFile()
```

Return the resource file containing the downloaded resource. This method should only be called if the resource is loaded and `getResourceLoadType()` returns `LOAD_TYPE_FILE`.

The provider must ensure that the returned file is readable by the GEM implementation.

Returns:

File - the resource load file.

Throws:

`java.lang.IllegalStateException` - if the resource is not loaded or is loaded to a byte array in memory.

getResourceLoadType()

```
public int getResourceLoadType()
```

Return a value indicating where the associated resource has been downloaded. Possible values include:

- `LOAD_TYPE_MEMORY` - indicates that the resource was loaded into memory as a byte array, and can be retrieved using the `getResourceBytes()` method. `LOAD_TYPE_FILE` - indicates that the resource was loaded to a file on a local physical file system, and can be retrieved using the `getResourceFile()` method.

Returns:

int - the load type.

Throws:

`java.lang.IllegalStateException` - if the resource is not loaded.

getURI()

```
public java.lang.String getURI()
```

Return the URI identifying the resource to be downloaded.

Returns:

String the URI for this resource.

isLoaded()

```
public boolean isLoaded()
```

Return a flag indicating whether the resource has been succesfully downloaded.

isMetadataLoaded()

```
public boolean isMetadataLoaded()
```

Return a flag indicating whether the metadata for this resource (existence flag, content length, and MIME type) is available for retrieval via the corresponding accessor methods. Depending on the specifics of the provider implementation and transport mechanism, it may be that this information is available without needing to call the `loadMetadata` method; that method should be called only when this method returns `false`.

Returns:

boolean - Flag indicating whether the metadata for this resource has been loaded and is ready for retrieval via relevant accessor methods.

loadMetadata(AsyncResourceLoadListener)

```
public void loadMetadata(org.dvb.spi.ict.AsyncResourceLoadListener listener)
throws ResourceLoadException
```


Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 798 of 861

Load the resource metadata (existence flag, content length, and MIME type) for this resource. Note that some providers will be able to determine the metadata information without downloading the resource content by using cached directory lookup information, but in general this may not be the case. In cases where the provider must download the resource to find the directory information, this call may have an equivalent effect to calling `asynchronousLoad()`. If a load request is already in progress, this method may also block until the outstanding load request is complete.

This method may fail either asynchronously with notification event sent to the listener or else synchronously during this method call.

Once the metadata is loaded, the following methods may be called:

- `exists()`
- `getMimeType()`
- `getLength()`

Parameters:

`listener` - A listener to be notified of asynchronous loading events.

Throws:

`ResourceLoadException`

removeUpdateListener(ResourceUpdateListener)

```
public void removeUpdateListener(org.dvb.spi.ict.ResourceUpdateListener listener)
```

Remove a listener that is currently receiving notifications of updates to this resource. If the specified listener has not previously been added, then calling this method has no effect.

Parameters:

`listener` - the listener to be removed.

synchronousLoad()

```
public void synchronousLoad()
throws ResourceLoadException
```

Load the resource. When this method returns, the resource will have been succesfully downloaded. If the download fails, then a ResourceLoadException will be thrown.

Throws:

`ResourceLoadException`

unload()

```
public void unload()
```

Tells the service provider that the downloaded resource object is no longer used, and therefore any resources allocated to this object can be freed. This method puts the ResourceTransportObject back into the unloaded state.

Throws:

`java.lang.IllegalStateException` - if the resource is not loaded.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 799 of 861

org.dvb.spi.ict

ResourceUpdateListener

Declaration

```
public interface ResourceUpdateListener
```

Description

This interface will be implemented by listeners that are part of the GEM implementation so that they are notified of updates to resources downloaded by an interaction channel tranport service provider.

Note that in the case where a DVB-J application registers an ObjectChangeEventListener on a DSMCCObject that refers to an interaction channel file system file, the GEM implementation will be respsonsible for mapping update method calls from this method to calls to the ObjectChangeEventListener (i.e. applications will not be able to reference the service provider classe directly).

Methods

resourceObjectUpdated(ResourceObjectUpdateEvent)

```
public void resourceObjectUpdated(org.dvb.spi.ict.ResourceObjectUpdateEvent e)
```

Notify the listener that a downloaded resource object has been updated. Details of the update are described in the specified event object.

Parameters:

`e` - the event object describing the change.

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

Annex AO (normative): Services and the service list

AO.1 Service list creation

The first entry point for services is the SIManager object. This object is responsible for aggregating services and delivering filtered lists of available services. It is assumed that when this object is instantiated it immediately begins to build the list of available services e.g. via the SD&S process to discover available Service Providers and their Service Offerings. The inputs to the list of available services shall be as follows:

- One input to the list of available services shall be all the services listed in the broadcast discovery information records for the selected service provider(s) - both "TS Full SI" and "TS - Optional SI". At least one of the service providers signalled in the service provider discovery records shall be selected. Support for simultaneously selecting more than one such service provider is optional however if more than one such service provider is selected simultaneously then services from all selected providers shall be used as input to the list of available services. Hence if the same service is available from multiple selected service providers then they shall appear in the service list as a single transport independent service.

- On GEM terminals supporting both classic DVB and IP network interfaces, a second input to the list of available services shall be the list discovered by the classical service scan (see clause Z.5, "Multiple service contexts in a GEM platform" and MHP [1], clause O.2.1, "javax.tv.service.Service").

- A third input to the list of available services shall be those available through currently registered selection providers (org.dvb.spi.selection.SelectionProvider).

- If services are discovered via the broadband content guide, these should not form an additional input to the list of available services.

The inputs to the list of available services shall be combined as follows:

- If more than one input contains the same service (identified by textual service identifier), the services shall be combined into a single entry in the available services list - even if they have different original_network_id and/or service_id.

- If more than one input contains the same service (identified by original network identifier and service identifier), the services shall be combined into a single entry in the available services list. The transport_stream_id shall be ignored for the purposes of this comparison. This comparison shall only happen between services without a textual_service_identifier.

AO.2 SIManager

AO.2.1 Introduction (informative)

The SIManager provides a single view over all the available SI sources for the GEM terminal. On GEM terminals supporting both classic DVB and IP network interfaces, information from both sources is merged into this single view.

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

AO.2.2 Method definition

AO.2.2.1 Methods without modified or extended semantics

The following list of methods have no modifications or extensions to their semantics defined by GEM.

- filterServices.
- getService.
- getRatingDimension.
- retrieveServiceDetails.
- registerInterest.
- retrieveProgramEvent.
- retrieveSIElement.

These methods shall support references to IP delivered content and services according to their specification.

For the methods retrieveProgramEvent and retrieveSIElement, on GEM terminals where BCG is not supported, where information from BCG is required in order to achieve the specified behaviour, the method call shall fail as specified for that method when that information is not available - even if that information would be available to GEM terminals supporting BCG.

AO.2.2.2 getTransports

The getTransports method shall return a Transport object for the broadband (IPTV or OTT) content delivery mechanism which implements BouquetCollection, NetworkCollection and TransportStreamCollection. This is in addition to Transport objects for those classical DVB content delivery mechanisms included in the GEM terminal. The Transport object for IPTV/OTT shall be as follows:

- The BouquetCollection shall include all bouquets defined in the package discovery information record.
- The TransportStreamCollection is not required to be implemented.
- The NetworkCollection is not required to be implemented.

AO.2.2.3 getSupportedDimensions

GEM does not require any rating dimensions to be available for IPTV/OTT delivered content.

See MHP [1], clause O.2.6.1 for the behaviour of this method in GEM terminals including a classical DVB network interface.

AO.3 Services

This clause considers four different types of services:

- Services only signalled in SD&S (in the TS Full SI Broadcast Discovery Information Record) where the information about the service comes from the stream itself. These will be both transport independent and transport dependent.
- Services only signalled in SD&S (in the TS Partial SI Broadcast Discovery Information Record) where the information about the service comes from the SD&S entry for the service (except as defined by the primary SI source attribute - see clause AO.3.2.1, "General"). These will be both transport independent and transport dependent.

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

- Services only signalled in DVB-SI which shall be supported as defined in MHP [1], clause O.2.1, "javax.tv.service.Service".
- Services signalled in more than one mechanism which by definition are only transport independent.

A fifth type of service is found in GEM terminals supporting the broadband content guide option as defined in clause 15, "Detailed platform profile definitions". These are services signalled in SD&S (in the TS Partial SI Broadcast Discovery Information Record) for which information is also available via the BCG. These are addressed in more detail in clause AQ.2 "javax.tv.service.Service and javax.tv.service.navigation.ServiceDetails".

AO.3.1 Services only signalled in TS Full SI Broadcast Discovery Information Record

AO.3.1.1 getLocator

When called on a transport independent service, the getLocator method shall return a transport independent locator for the service as defined in clause 14.9, "Content referencing for IPTV/ OTT".

When called on a transport dependent service, the getLocator method shall return a transport dependent locator for the service as defined in clause 14.9, "Content referencing for IPTV/ OTT".

AO.3.1.2 getName

The getName method shall return the name as defined in MHP [1], clause O.2.1.1, "getName". If no name is available (e.g. the service concerned has never been received by the GEM terminal) then a string of length zero shall be returned.

AO.3.1.3 getServiceType

The getServiceType method shall return the value defined in MHP [1], clause O.2.1.2, "getServiceType". If no type is available (e.g. the service concerned has never been received by the GEM terminal) then UNKNOWN shall be returned.

AO.3.1.4 retrieveDetails

When the service concerned is being received by the GEM terminal, the retrieveDetails method shall retrieve the ServiceDetails as defined in AP.4.1, "javax.tv.service.navigation.ServiceDetails". Otherwise this method shall behave as defined in clause AP.2.2, "Information available only when a stream is actually received".

AO.3.1.5 hasMultipleInstances

The hasMultipleInstances method shall return true if and only if the service's entry in the broadcast discovery information record includes both IGMP and RTSP protocols.

AO.3.2 Services only signalled in TS Partial SI Broadcast Discovery Information Record

AO.3.2.1 General

Implementations of the Java TV APIs shall ignore DVB-SI tables if present in services signalled with TS Partial SI Broadcast Discovery Information records. There is only a single source of SI for such services and that is SD&S.

NOTE 1: The "primary SI source" attribute is ignored.

NOTE 2: Applications wishing to access any DVB-SI information in such services may do this using the org.dvb.si package.

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

AO.3.2.2 getLocator

When called on a transport independent service, the getLocator method shall return a transport independent locator for the service as defined in clause 14.9, "Content referencing for IPTV/ OTT".

When called on a transport dependent service, the getLocator method shall return a transport dependent locator for the service as defined in clause 14.9, "Content referencing for IPTV/ OTT".

AO.3.2.3 getName

When called on a transport independent service, the getName method returns the name of the service as stored in the GEM terminal. Depending on the GEM terminal implementation, the end user may have the possibility to edit these names according to his preferences. In the absence of editing by the end user, one of the multilingual service names as defined in clause AP.2.1, "dvb:MultilingualType" shall be returned.

When called on a transport dependent service, one of the multilingual service names as defined in clause AP.2.1, "dvb:MultilingualType" shall be returned.

AO.3.2.4 getServiceType

The getServiceType method shall return the service type signalled in the discovery information record mapped into Java as defined in MHP [1], clause O.2.3, "javax.tv.service.ServiceType".

AO.3.2.5 retrieveDetails

The retrieveDetails method shall return a ServiceDetails instance containing information from the discovery information record as described in clause AP.5.1, "javax.tv.service.navigation.ServiceDetails".

AO.3.2.6 hasMultipleInstances

The hasMultipleInstances method shall return true if and only if the service's entry in the broadcast discovery information record includes both IGMP and RTSP protocols.

AO.3.3 Services signalled in more than one mechanism

These services are always transport independent by definition. When a transport independent service of this type is transformed into transport dependent services, those transport dependent services.

AO.3.3.1 getLocator

If all instances of the service are identified by the same textual service identifier then the method getLocator shall return a transport independent dvb: locator including that textual service identifier otherwise it shall return a transport independent dvb: locator with the DVB triplet.

The following apply to the transport independent locator returned from this method:

- The Locator.hasMultipleTransformations method shall return true.
- The LocatorFactory.transformLocator method shall return one transport dependent locator for each actual service instance.

AO.3.3.2 getName

The getName method returns the name of the service as stored in the GEM terminal. Depending on the GEM terminal implementation, the end user may have the possibility to edit these names according to his preferences. In the absence of editing by the end user, an implementation specific choice shall be made of one of between the mechanisms in which the service is signalled. The name shall be returned according to the specification for the chosen mechanism - MHP [1], clause O.2.1.1, "getName" or one of the multilingual service names as defined in clause AP.2.1, "dvb:MultilingualType".

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

AO.3.3.3 getServiceType

An implementation specific choice shall be made of one of between the mechanisms in which the service is signalled. The type returned shall be according to the specification for the chosen mechanism - MHP [1], clause O.2.1.2, "getServiceType" or clause AO.3.2.4, "getServiceType".

AO.3.3.4 retrieveDetails

This method shall behave as specified.

AO.3.3.5 hasMultipleInstances

The hasMultipleInstances method shall return true.

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

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

AP.1 Introduction (informative)

The following clause defines the mapping between the Java TV SI API and the service discovery and selection. This section applies to services signalled via SD&S. It is divided into 5 clauses:

- generic issues which apply to multiple methods in the API;
- mappings which apply to services regardless of whether they are signalled via TS Full SI Broadcast Discovery Information Record or the TS Partial SI Broadcast Discovery Information Record;
- mappings which only apply to services signalled via TS Full SI Broadcast Discovery Information Record;
- mappings which only apply to services signalled via TS Partial SI Broadcast Discovery Information Record.

AP.2 Generic issues

AP.2.1 dvb:MultilingualType

For fields whose value is signalled via a MultilingualType element, if the language returned by javax.tv.service.SIManager.getPreferredLanguage corresponds to a signalled language then the text description in that language shall be used. Otherwise an implementation dependent selection between the available languages shall be made.

AP.2.2 Information available only when a stream is actually received

The following shall apply for information used in this mapping that is only guaranteed to be available when the GEM terminal is receiving the stream concerned.

- When the service concerned is being received by the GEM terminal, the information shall be returned from the received service.
- When the service concerned is not being received by the GEM terminal, the information may be returned if the GEM terminal has it in cache from when the service was previously received.
- If the service is not being received and the information is not cached, the request shall fail as defined for the method concerned when data is unavailable. (This means an SIRequestFailureType of DATA_UNAVAILABLE for methods passed an SIRequestor). In this case, the GEM terminal shall not request the network to supply the service concerned.

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

AP.3 Mappings independent of Discovery Information Record Type

AP.3.1 javax.tv.service.navigation.ServiceProviderInformation

The name of the service provider shall be retrieved from the multilingual service provider name attribute of the service provider discovery record for the service provider. The name returned shall be selected from those available as specified by clause AP.2.1, "dvb:MultilingualType".

AP.3.2 javax.tv.service.navigation.StreamType

When the service concerned is being received by the GEM terminal, the stream type shall be as defined in MHP [1], clause O.2.4. When the service concerned is not being received by the GEM terminal but information on the stream types of the components of that service are cached in the GEM terminal then the cached data shall be returned. Otherwise StreamType.UNKNOWN shall be returned.

AP.3.3 javax.tv.service.navigation.ServiceComponent

This shall be supported as defined in MHP [1], clause O.2.2, "javax.tv.service.navigation.ServiceComponent".

AP.3.4 javax.tv.service.transport.Network

This is not required to be implemented but if implemented, shall be supported as defined in MHP [1], clause O.2.14, "javax.tv.service.transport.Network".

NOTE: This is not required to be implemented because NetworkCollection is not required to be implemented.

AP.3.5 javax.tv.service.transport.Transport

AP.3.5.1 getDeliverySystemType

The getDeliverySystemType method shall return the delivery system type defined in org.dvb.service.navigation.IPDeliverySystemType.

AP.3.6 javax.tv.service.transport.Bouquet

Each package in the Package Discovery Information record shall be represented as an instance of Bouquet.

AP.3.6.1 getBouquetID

The getBouquetID method shall return the package id from the Package Discovery Information record.

AP.3.6.2 getName

The getName method shall return one of the multilingual package names selected as defined in clause AP.2.1, "dvb:MultilingualType".

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

AP.3.7 javax.tv.service.transport.BouquetCollection

There is no requirement to monitor for changes more frequently than once per day. When changes are detected, BouquetChangeEvents will be generated when packages are added or removed. BouquetChangeEvents with SIChangeType MODIFY shall only be generated when it is detected that the contents of the service list for a package changes.

AP.3.8 javax.tv.service.transport.TransportStream

This shall be supported as defined in MHP [1], clause O.2.15, "javax.tv.service.transport.TransportStream" except that the transport stream ID returned shall always be taken from the broadcast discovery information record.

AP.3.9 javax.tv.service.transport.TransportStreamCollection

This is not required to be implemented. If it is implemented, there is no requirement to monitor for changes more frequently than once per day.

AP.3.10 javax.tv.service.transport.NetworkCollection

This is not required to be implemented. If it is implemented, there is no requirement to monitor for changes more frequently than once per day.

NOTE: Where broadcast services are being relayed for carriage over an IP network, it is unclear whether any re-writing of DVB-SI network_ids will be done. Hence the network_ids may be more varied than would be the case in a classical DVB environment.

AP.3.11 javax.tv.service.ServiceInformationType

The present document defines additional service information types in org.dvb.service.IPServiceInformationType.

AP.3.12 javax.tv.service.ServiceType

The translation from DVB-SI encoding shall be as defined in MHP [1], clause O.2.3, "javax.tv.service.ServiceType".

AP.3.13 javax.tv.service.navigation.DeliverySystemType

The delivery system type returned shall be org.dvb.service.navigation.IPDeliverySystemType.IP as defined in the present document.

AP.3.14 ServiceNumber

Support for ServiceNumber is not defined in GEM.

AP.3.15 javax.tv.service.navigation.CAIdentification

Support for CAIdentification is not defined in GEM.

AP.3.16 javax.tv.service.navigation.FavoriteServicesName

This is not a signalling issue but is an attribute of the navigator UI.

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

AP.3.17 javax.tv.service.transport.Network

This is not required to be implemented but if implemented, shall be supported as defined in MHP [1], clause O.2.14, "javax.tv.service.transport.Network".

AP.4 Mappings specific to TS Full SI Broadcast Discovery Information Record

AP.4.1 javax.tv.service.navigation.ServiceDetails

These shall be supported as defined in MHP [1], clause O.2.8, "javax.tv.service.navigation.ServiceDetails".

When the service concerned is being received by the GEM terminal, the GEM terminal shall monitor the SI of the service and post events to ServiceComponentChangeListeners as specified. When it is not being received, no SI monitoring is possible and hence no events will be posted.

Information on the components of a service shall be supported as defined by clause AP.2.2, "Information available only when a stream is actually received" Information available only when a stream is actually received.

ServiceDetails objects shall implement the ServiceProviderInformation interface.

The Locator returned by getLocator shall be a transport dependent locator for this specific service.

AP.5 Mappings specific to TS Partial SI Broadcast Discovery Information Record

AP.5.1 javax.tv.service.navigation.ServiceDetails

ServiceDetails objects shall implement the ServiceProviderInformation interface.

AP.5.1.1 getDeliverySystemType

The delivery system type returned shall be org.dvb.service.navigation.IPDeliverySystemType.IP as defined in the present document.

AP.5.1.2 getLongName

The getLongName method shall return one of the multilingual service names as defined in clause AP.2.1, "dvb:MultilingualType".

AP.5.1.3 getService

The getService method shall return the appropriate service object for this service details.

AP.5.1.4 getLocator

The getLocator method shall return a transport dependent locator for this specific service as defined in clause 14.9, "Content referencing for IPTV/ OTT".

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

AP.5.1.5 getServiceType

The getServiceType method shall return the service type signalled in the discovery information record decoded according to MHP [1], clause O.2.3, "javax.tv.service.ServiceType".

AP.5.1.6 retrieveComponents

Information on the service components shall be returned based on the signalling in the PMT of the transport stream carrying the service. Clause AP.2.2, "Information available only when a stream is actually received" Information available only when a stream is actually received shall apply.

AP.5.1.7 retrieveServiceDescription

The retrieveServiceDescription method shall return one of the optional multilingual service descriptions as defined by clause AP.2.1, "dvb:MultilingualType" or fail with an SIRequestFailureType of DATA_UNAVAILABLE if none of those descriptions are present. Clause AP.2.2, "Information available only when a stream is actually received". Information available only when a stream is actually received shall apply.

AP.5.1.8 addServiceComponentChangeListener

If the service concerned is currently being received by the GEM terminal then the PMT of the transport stream carrying the service shall be monitored for changes to service components. Otherwise monitoring the SI for changes to service components is impossible.

AP.5.2 javax.tv.service.navigation.ServiceDescription

The service description shall be one of the descriptions in the Description element of the SI element. The choice of which description to return shall be as defined by clause AP.2.1, "dvb:MultilingualType".

AP.5.3 javax.tv.service.navigation.ServiceComponent

AP.5.3.1 getName

The getName method shall return an empty string

AP.5.3.2 getAssociatedLanguage

The getAssociatedLanguage method shall return an empty string

AP.5.3.3 getStreamType

The getStreamType method shall return a StreamType as defined in clause AP.3.2, "javax.tv.service.navigation.StreamType".

AP.5.3.4 getService

The getService method shall return the appropriate service.

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

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

The following clauses define the mapping between the Java TV SI API and the TV-Anytime metadata specified by the broadband content guide - TS 102 539 [81].

AQ.1 Finding the BCG Provider

The Java TV API shall use the SD&S Service(s)DescriptionLocation field to locate the BCG provider to use. The SD&S information may specify a BCG provider for a range of services or one per service.

If present in the SD&S information the "preferred" attribute shall be used to identify which BCG provider takes precedence. If not present then GEM does not define which of the signalled providers to use.

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

In GEM terminals supporting the broadband content guide, clause 6.6 of TS 102 539 [81] shall apply.

NOTE: This means information from the BCG takes precedence over information from SD&S. Information from SD&S is only used when fields which are optional in BCG are not present.

AQ.3 javax.tv.service.guide.ProgramSchedule

This maps to a collection of <BroadcastEvent> and <ScheduleEvent> elements in the <ProgramLocationTable> for a particular service. <ScheduleEvent> elements are contained within one to many <Schedule> elements.

AQ.3.1 retrieveCurrentProgramEvent

The retrieveCurrentProgramEvent method shall return a ProgramEvent based on the instance of <ScheduleEvent> or <BroadcastEvent> that has <StartTime> and <EndTime> overlapping the current time.

AQ.3.2 retrieveFutureProgramEvents

The retrieveFutureProgramEvents method shall return all ProgramEvents based on instances of <ScheduleEvent> or <BroadcastEvent> that have <StartTime> and <EndTime> between the specified begin and end time.

AQ.3.3 retrieveNextProgramEvent

The retrieveNextProgramEvent method shall return a ProgramEvent based on an instance of <ScheduleEvent> or <BroadcastEvent> that follows the current event in the schedule in terms of <StartTime>.

AQ.3.4 retrieveProgramEvent

The retrieveProgramEvent method shall return the ProgramEvent that matches the Locator as defined by clause 11.7.11, "TV-Anytime content referencing and metadata".

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

AQ.3.5 addListener

No additional semantics are defined in the present document.

NOTE 1: TV-Anytime bi-directional updates are based on polling.

NOTE 2: <ScheduleEvent> elements can only be updated in TV-Anytime as part of a whole <Schedule> element, whereas <BroadcastEvent> elements can be updated independently.

AQ.3.6 removeListener

No additional semantics are defined in the present document.

AQ.3.7 getServiceLocator

The getServiceLocator method shall return a Locator based on the TV-Anytime <ServiceURL> obtained from the Service Information Table entry which matches the serviceId of the <Schedule> or <BroadcastEvent> element.

AQ.4 javax.tv.service.guide.ProgramEvent

ProgramEvent shall map either to a TV-Anytime <ScheduleEvent> or to a <BroadcastEvent> element within the <ProgramLocationTable>.

AQ.4.1 getLocator

The getLocator method shall return a Locator for the ProgramEvent obtained by resolving the CRID of the <ScheduleEvent> or <BroadcastEvent>. This locator shall comply with clause 11.7.11, "TV-Anytime content referencing and metadata".

AQ.4.2 getStartTime

The getStartTime method shall return the time specified in the <PublishedStartTime> of a <ScheduleEvent> or <BroadcastEvent> in the <ProgramLocationTable>.

AQ.4.3 getEndTime

The getEndTime method shall return the time specified in the <PublishedEndTime> of a <ScheduleEvent> or <BroadcastEvent> element in the <ProgramLocationTable>. If not present then the <PublishedStartTime> and <PublishedDuration> shall be used to calculate this.

AQ.4.4 getDuration

The getDuration method shall return the time specified in the <PublishedDuration> of a <ScheduleEvent> or <BroadcastEvent> element in the <ProgramLocationTable>.

If <PublishedDuration> is not present then the <PublishedStartTime> and <PublishedEndTime> shall be used to calculate this.

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

AQ.4.5 getName

The getName method shall return the String specified in the <Title> element of <InstanceDescription>, contained in either a <ScheduleEvent> or <BroadcastEvent> element in the <ProgramLocationTable>.

If <InstanceDescription> is not present then the <Title> element from the <BasicContentDescription> element in the <ProgramInformationTable> shall be returned

AQ.4.6 retrieveDescription

The retrieveDescription method shall return the String specified in the short <Synopsis> element of <InstanceDescription>, contained in either a <ScheduleEvent> or <BroadcastEvent> element in the <ProgramLocationTable>.

If <InstanceDescription> is not present then the short <Synopsis> element from the <BasicContentDescription> element in the <ProgramInformationTable> shall be used.

If the short <Synopsis> is not present then the long <Synopsis> shall be used.

AQ.4.7 getRating

The getRating method shall return a ContentRatingAdvisory based on the TV-Anytime <ParentalGuidance> element from the <ContentDescription> element contained in the <ProgramInformationTable>.

The ContenRatingAdvisory shall only represent ratings that are part of the rating dimensions returned by the method SIManager.getSupportedDimensions.

If there is no TV-Anytime rating information present within the supported dimensions then this method shall return null.

AQ.4.8 getService

No additional semantics are defined in GEM.

AQ.4.9 retrieveComponents

No mapping is defined in GEM.

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

The ProgramEventDescription shall be constructed from the short <Synopsis> element of <InstanceDescription>, contained in either a <ScheduleEvent> or <BroadcastEvent> element in the <ProgramLocationTable>.

If <InstanceDescription> is not present then the short <Synopsis> element from the <BasicContentDescription> element in the <ProgramInformationTable> shall be used.

If the short <Synopsis> is not present then the long <Synopsis> shall be used.

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

Annex AR (normative): XML encoding for AIT

Signaling of external applications is optional for IPTV targets.

AR.1 Introduction

This annex defines an XML encoding for the AIT in addition to the already existing MPEG-2 table and section based encoding defined in MHP [1], clause 10, "Application Signalling". Since the first intended use for this XML encoding is in conjunction with the SD&S defined in TS 102 034 [80], this encoding follows the same format and re-uses already defined elements and types.

The following features of the MPEG-2 encoding are not supported in the XML encoding:

- Application types different from DVB-J and DVB-HTML are not currently taken into account.
- DVB-HTML specific descriptors are not defined in GEM.
- Signalling of applications transported in DSM-CC object carousel is not supported.

The semantics of the fields defined in this annex shall be identical to those of the corresponding fields in the existing MPEG-2 table and section based encoding as defined by GEM and extended by OCAP [5].

AR.2 Extensions to defined SD&S elements

AR.2.1 Package

Add to Package, an optional list of applications and an optional list of externally authorised applications:

```
<xsd:element name="applicationList" type="dvb:mhp:ApplicationList" minOccurs="0"/>

<xsd:element name="ExternalapplicationAuthorisedList"
type="dvb:mhp:ExternalApplicationIdentifier**" minOccurs="0"/>

<xsd:complexType name="PackagedServices">
    <xsd:complexContent>
        <xsd:extension base="dvb:OfferingBase">
            <xsd:sequence>
                <xsd:element name="Package" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="PackageName" type="dvb:MultilingualType"
                            maxOccurs="unbounded"/>
                            <xsd:element name="IPService" maxOccurs="unbounded">
                                <xsd:complexType>
                                    <xsd:sequence>
                                        <xsd:element name="TextualID" type="dvb:TextualIdentifier"/>
                                        <xsd:element name="DescriptionLocation"
type="dvb:DescriptionLocation"
                                        minOccurs="0"/>
                                    </xsd:sequence>
                                </xsd:complexType>
                            </xsd:element>
                            <xsd:element name="CountryAvailability" type="dvb:CountryAvailability"
                            minOccurs="0" maxOccurs="unbounded"/>
                            <xsd:element name="PackageDescription" type="dvb:DescriptionLocation"
                            minOccurs="0" maxOccurs="unbounded"/>
                            <xsd:element name="applicationList" type="dvb:mhp:ApplicationList"
                            minOccurs="0"/>
                            <xsd:element name="ExternalapplicationAuthorisedList"
                            type="dvb:mhp:ExternalApplicationIdentifier" minOccurs="0"/>
                        </xsd:sequence>
```


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

```
                    <xsd:attribute name="Id" type="dvb:Hexadecimal16bit" use="required"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

The application list signalled in the package is aimed to contain the whole set of applications which are available for all the IPServices signalled in such package.

AR.2.2 IP Service

Add the following elements, as optional:

```
<xsd:element name="applicationList" type="dvbmhp:ApplicationList" minOccurs="0"/>
<xsd:element name="ExternalapplicationAuthorisedList"
type="dvb:ExternalApplicationIdentifier" minOccurs="0"/>
```

Therefore the IPService definition is modified as follows:

```
<xsd:complexType name="IPService">
    <xsd:sequence>
        <xsd:element name="ServiceLocation" type="dvb:ServiceLocation"/>
        <xsd:element name="TextualIdentifier" type="dvb:TextualIdentifier"/>
        <xsd:element name="DVBTriplet" type="dvb:DVBTriplet"/>
        <xsd:element name="MaxBitrate" type="xsd:positiveInteger" minOccurs="0"/>
        <xsd:element name="SI" type="dvb:SI" minOccurs="0"/>
        <xsd:element name="applicationList" type="dvb:MHPApplicationList" minOccurs="0"/>
        <xsd:element name="ExternalapplicationAuthorisedList"
        type="dvb:ExternalApplicationIdentifier" minOccurs="0"/>
        <xsd:element name="VideoCoding" type="dvb:VideoCoding" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="AudioCoding" type="dvb:AudioCoding" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

AR.3 New XML element definitions

AR.3.1 ApplicationList

```
<xsd:complexType name="ApplicationList">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="application" type="dvb:mhp:Application"/>
    </xsd:sequence>
</xsd:complexType>
```

AR.3.2 Application

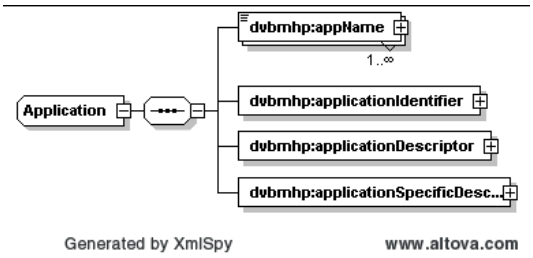
```
<xsd:complexType name="Application">
        <xsd:sequence>
            <xsd:element name="appName" type="dvb:MultilingualType" maxOccurs="unbounded"/>
            <xsd:element name="applicationIdentifier" type="dvb:mhp:ApplicationIdentifier"/>
            <xsd:element name="applicationDescriptor" type="dvb:mhp:ApplicationDescriptor"/>
            <xsd:element name="applicationSpecificDescriptor"
            type="dvb:mhp:ApplicationSpecificDescriptor"/>
            <xsd:element name="providerExportDescriptor"
            type="dvb:mhp:exportDescriptor"/>
            <xsd:element name="providerUsageDescriptor"
            type="dvb:mhp:usageDescriptor"/>
        </xsd:sequence>
</xsd:complexType>
```

A GEM application can be completely described by:

- An Application Name which can be multilingual. (appName).
- An Unique identification (applicationIdentifier).

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

- A generic descriptor which is common and mandatory for both types of applications (DVB-J & DVB-HTML) defined in the present document. This generic descriptor is valid also for other type of applications, e.g OCAP.
- An Application Specific descriptor which will depend upon the signalled application, e.g. DVB-J, DVB-HTML.

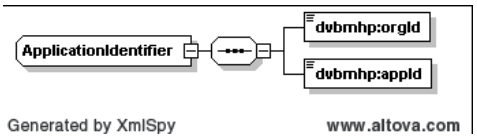


AR.3.3 ApplicationIdentifier

```
<xsd:complexType name="ApplicationIdentifier">
    <xsd:sequence>
        <xsd:element name="orgId" type="xsd:unsignedInt"/>
        <xsd:element name="appId" type="xsd:unsignedShort"/>
    </xsd:sequence>
</xsd:complexType>
```

As defined in clause 10.4.3, "Content of the Application Description" of the present document, an application is uniquely identified by:

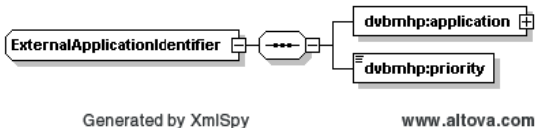
- OrgId, organisation identifier, it is a globally unique value identifying the organisation that is responsible for the application.
- AppId, application identifier, it is allocated by the organisation registered with the organisation identifier who decides the policy for allocation within the organisation.



AR.3.4 ExternalApplicationIdentifier

```
<xsd:complexType name="ExternalApplicationIdentifier">
    <xsd:sequence>
        <xsd:element name="application" type="dvb:mhp:ApplicationIdentifier"/>
        <xsd:element name="priority" type="dvb:Hexadecimal8bit"/>
    </xsd:sequence>
</xsd:complexType>
```

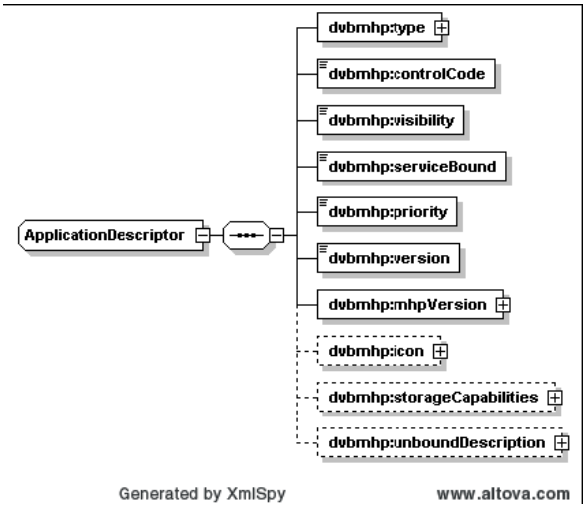
As defined in clause 10.7.5, "External application authorization descriptor" of MHP [1], an external application Identifier is used to signal an application which is allowed to continue running in the current service even if it is not signalled in it.



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

AR.3.5 ApplicationDescriptor

```
<xsd:complexType name="ApplicationDescriptor">
    <xsd:sequence>
        <xsd:element name="type" type="dvb:mhp:ApplicationType"/>
        <xsd:element name="controlCode" type="dvb:mhp:ApplicationControlCode"/>
        <xsd:element name="visibility" type="dvb:mhp:VisibilityDescriptor"/>
        <xsd:element name="serviceBound" type="xsd:boolean" default="true"/>
        <xsd:element name="priority" type="dvb:Hexadecimal8bit"/>
        <xsd:element name="version" type="dvb:Version"/>
        <xsd:element name="mhpVersion" type="dvb:mhp:MhpVersion"/>
        <xsd:element name="icon" type="dvb:mhp:IconDescriptor" minOccurs="0"/>
        <xsd:element name="storageCapabilities" type="dvb:mhp:StorageCapabilities" minOccurs="0"/>
        <xsd:element name="unboundDescription" type="dvb:mhp:UnboundApplicationDescriptor"
        minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```



This contents of this complex type are mostly those defined in clause 10.7.3, "Application descriptor" of MHP [1].

The simple type elements are defined as follows:

visibility: this element specifies whether the application is suitable to be offered to the end-user for them to decide if the application should be launched. Therefore "false" value means that 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., whereas a "true" value means that application shall not be visible to users but shall be visible to applications via an application listing API.

serviceBound: whether the application is bound to a service or not as defined by the `service_bound_flag`.

priority: This field identifies a relative priority between the applications signalled in this service.

- 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 may make an implementation-dependent choice on which to start.

version: Version of the application which 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 reused, in the event that the number range is exhausted a new application id shall be used.

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

mhpVersion: version of the GEM as defined in 10.7.3 "Application descriptor" of MHP [1].

icon: to bind an icon to the application.

storageCapabilities: this optional element shall be added for giving the receiver the required information to store / cache the application.

unboundDescription: this descriptor only shall be used when serviceBound is set to false, i.e. it is an unbound application.

AR.3.6 VisibilityDescriptor

```
<xsd:simpleType name="VisibilityDescriptor">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="NOT_VISIBLE_ALL"/>
        <xsd:enumeration value="NOT_VISIBLE_USERS "/>
        <xsd:enumeration value="VISIBLE_ALL"/>
    </xsd:restriction>
</xsd:simpleType>
```

The following correspond to the values of the field defined in MHP [1], table 84, "Definition of visibility states for applications".

NOT_VISIBLE_ALWAYS: 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.

NOT_VISIBLE_USERS: This application shall not be visible to users but shall be visible to applications via an application listing API.

VISIBLE_ALL: This application can be visible to users and shall be visible to applications via the application listing API.

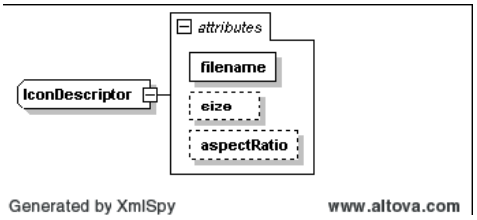
AR.3.7 IconDescriptor

```
<xsd:complexType name="IconDescriptor">
    <xsd:attribute name="filename" type="xsd:string" use="required"/>
    <xsd:attribute name="size" type="xsd:unsignedShort" use="optional"/>
    <xsd:attribute name="aspectRatio" type="dvb:AspectRatio" use="optional"/>
</xsd:complexType>
```

As defined in clause 10.7.4.2, "Application icons descriptor" of MHP [1] the icon Descriptor serves to signal the presence of an icon representing the application. Size and aspectRatio attributes are defined as optional since they can be determined as defined in MHP [1], table 88, "Definition of different icon flags".

E.g.

```
<icon filename="dvb.icon.1"/> , size = 32x32 pixel square
```



AR.3.8 AspectRatio

```
<xsd:simpleType name="AspectRatio">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="4_3"/>
        <xsd:enumeration value="16_9 "/>
    </xsd:restriction>
```


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

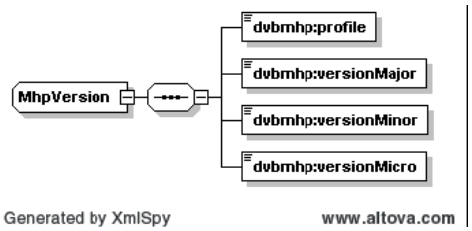
```
</xsd:simpleType>
```

These aspect ratios are the set of aspect ratios used in MHP [1], table 88, "Definition of different icon flags".

AR.3.9 MhpVersion

```
<xsd:complexType name="MhpVersion">
    <xsd:sequence minOccurs="1">
        <xsd:element name="profile" type=" dvb:Hexadecimal16bit "/>
        <xsd:element name="versionMajor" type=" dvb:Hexadecimal8bit "/>
        <xsd:element name="versionMinor" type=" dvb:Hexadecimal8bit "/>
        <xsd:element name="versionMicro" type=" dvb:Hexadecimal8bit "/>
    </xsd:sequence>
</xsd:simpleType>
```

See MHP [1] clause 10.7.3, "Application descriptor".



AR.3.10 StorageCapabilities

```
<xsd:complexType name="StorageCapabilities">
    <xsd:sequence minOccurs="0">
        <xsd:element name="storageProperty" type="dvb:StorageType"/>
        <xsd:element name="isStorable" type="xsd:boolean"/>
        <xsd:element name="canCache" type="xsd:boolean"/>
    </xsd:sequence>
    <xsd:attribute name="launchableFromBroadcast" type="xsd:boolean" use="required"/>
    <xsd:attribute name="launchableCompletelyFromCache" type="xsd:boolean" use="required"/>
    <xsd:attribute name="launchableWithOlderVersion" type="xsd:boolean" use="required"/>
</xsd:complexType>
```

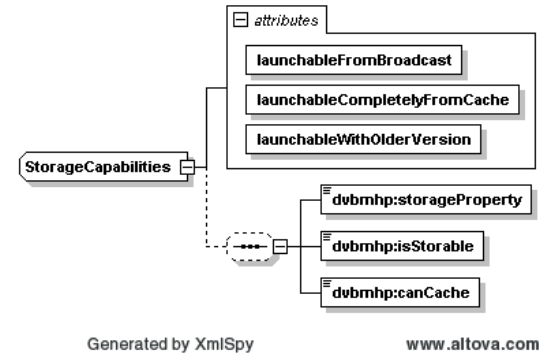
This descriptor, if present, serves to state whether the application can be stored or cached in the receiver as defined in MHP [1], clause 10.8.2, "Application storage descriptor".

isStorable: if set to true, it means that the application can be stored in the receiver.

canCache: if set to true, it means that this application can be run entirely from cache.

The attributes launchableFromBroadcast, launchableCompletelyFromCache, launchableWithOlderVersion have exactly the same meaning as the bit flags as defined in MHP [1], clause 10.8.2, "Application storage descriptor".

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



AR.3.11 StorageType

```
<xsd:simpleType name="StorageType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="BROADCAST-RELATED"/>
        <xsd:enumeration value="STANDALONE"/>
    </xsd:restriction>
</xsd:simpleType>
```

See MHP [1], table 113, "Semantics of storage property values".

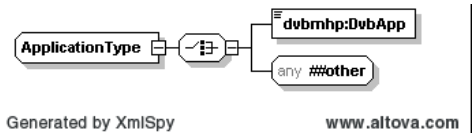
If the storage type is "BROADCAST-RELATED" then the application's life cycle is controlled by the broadcast service. Such applications are suitable for caching but not for running as a stand alone application.

If the storage type is "STANDALONE" then the application can usefully be launched by the user independently of a broadcast service.

AR.3.12 ApplicationType

```
<xsd:complexType name="ApplicationType">
        <xsd:choice>
            <xsd:element name="DvbApp" type="dvb:mhp:DvbApplicationType"/>
            <xsd:any namespace="##other" processContents="lax"/>
        </xsd:choice>
    </xsd:complexType>
```

See MHP [1], table 75, "Application types".



AR.3.13 DvbApplicationType

```
<xsd:simpleType name="DvbApplicationType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="DVB-J"/>
        <xsd:enumeration value="DVB-HTML"/>
    </xsd:restriction>
</xsd:simpleType>
```

This descriptor is a choice between the both application types as defined in MHP [1], table 75, "Application types".

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

AR.3.14 ApplicationControlCode

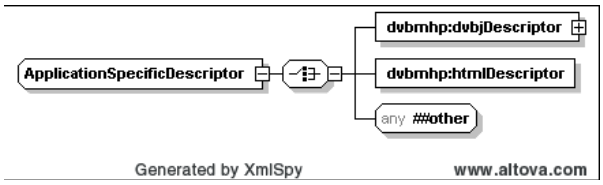
```
<xsd:simpleType name="ApplicationControlCode">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="AUTOSTART"/>
        <xsd:enumeration value="PRESENT"/>
        <xsd:enumeration value="DESTROY"/>
        <xsd:enumeration value="KILL"/>
        <xsd:enumeration value="REMOTE"/>
    </xsd:restriction>
</xsd:simpleType>
```

This descriptor serves to dynamically control application life cycle. The meaning of each one of the enumeration elements as well as the expected behaviour in the GEM receiver is fully defined in MHP [1], clause 10.7.3, "Application descriptor".

AR.3.15 ApplicationSpecificDescriptor

```
<xsd:complexType name="ApplicationSpecificDescriptor">
    <xsd:choice>
        <xsd:element name="dvbjDescriptor" type="dvb:mhp:DVBJDescriptor"/>
        <xsd:element name="htmlDescriptor" type="dvb:mhp:DVBHtmlDescriptor"/>
        <xsd:any namespace="##other" processContents="lax"/>
    </xsd:choice>
</xsd:complexType>
```

This descriptor contains the specific descriptor depending upon the type of application. As well as descriptors defined in the present document, it may also include descriptors from outside, e.g. a descriptor for OCAP applications.



AR.3.16 DVBJDescriptor

```
<xsd:complexType name="DVBJDescriptor">
    <xsd:sequence>
        <xsd:element name="location" type="dvb:DescriptionLocation"
        minOccurs="1" maxOccurs="unbounded"/>
        <xsd:element name="applicationStructure" type="dvb:mhp:ApplicationStructure"/>
    </xsd:sequence>
</xsd:complexType>
```

For executing DVB-J application, the receiver needs specifically where it is placed -its location- and how it is structured -in order to "rebuild" it-. The contents of this element correspond to the contents of MHP [1], table 97, "Syntax of selector bytes for interaction transport".

At least one location element must be provided in order to allow the receiver to find the ADF as well as the elements pointed by the ADF.

The location elements are of type dvb:DescriptionLocation which are defined in TS 102 034 [80] as URI. A set of URIs should be enough to find the elements. See example below:

e.g.

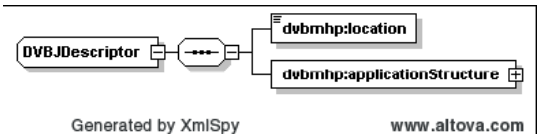
```
< DVBJDescriptor>
    < location>http://www.dvb.org/applications/application1.zip</location>
    < location>http://www.dvb.org/graphics</location>
    < location>http://www.dvb.org/shared/utils.zip</location>
    < location>http://www.ebu.ch/general/misc.zip</location>
    < location>udp://www.dvb.org/byudp </location>
    <applicationStructure>
            <!—Not necessary for this example-- >
```


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

```
<   / applicationStructure>
</ DVBJDescriptor >
```

Then the search order of for the "dvb.fontindex" file will be the following:

- In the contents of http://www.dvb.org/applications/application1.zip.
- From the URL http://www.dvb.org/graphics.
- In the contents of http://www.dvb.org/shared/utils.zip.
- In the contents of http://www.dvb.org/shared/misc.zip.
- In the URL udp://www.dvb.org/byudp.

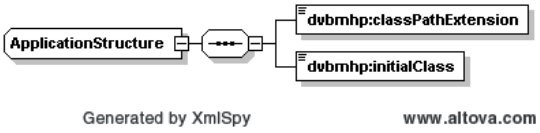


AR.3.17 ApplicationStructure

```
<xsd:complexType name="ApplicationStructure">
    <xsd:sequence>
        <xsd:element name="classPathExtension" type="xsd:string"/>
        <xsd:element name="initialClass" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
```

This complex type is an XML encoding of some of the information described in MHP [1], clause 10.9.2, "DVB-J application location descriptor".

In order to recreate the application, and execute it, only is required the set of files (class files, images, etc.) -ADF-, the classPathExtension, which can be "", and the initial class (i.e. the class which implements Xlet interface).



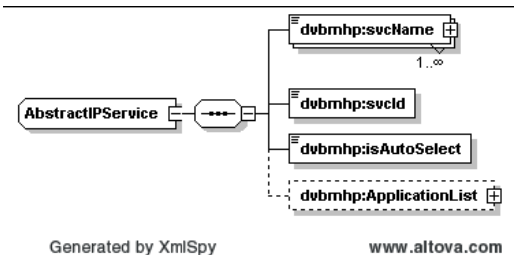
AR.3.18 AbstractIPService

Most of the information in this complex type is the same as that carried in the abstract service descriptor defined in OCAP [5].

```
<xsd:complexType name="AbstractIPService">
    <xsd:sequence>
        <xsd:element name="svcName" type="dvb:MultilingualType" maxOccurs="unbounded"/>
        <xsd:element name="svcId" type="dvb:mhp:Hexadecimal24bit"/>
        <xsd:element name="isAutoSelect" type="xsd:boolean"/>
        <xsd:element name="ApplicationList" type="dvb:mhp:ApplicationList" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="Hexadecimal24bit">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-9a-fA-F]{1}[1-9a-fA-F]{1}[0-9a-fA-F]{4}"/>
    </xsd:restriction>
</xsd:simpleType>
```


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

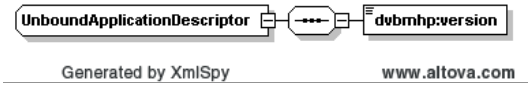


AR.3.19 UnboundApplicationDescriptor

Most of the information carried in this complex type is the same as that carried in the unbound application descriptor defined in OCAP [5].

```
<!-- To signal an unbound application -->
    <xsd:complexType name="UnboundApplicationDescriptor">
        <xsd:sequence>
            <xsd:element name="version" type="dvb:mhp:Hexadecimal32bit"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:simpleType name="Hexadecimal32bit">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="[0-9a-fA-F]{1,8}"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```



AR.3.20 Provider descriptors

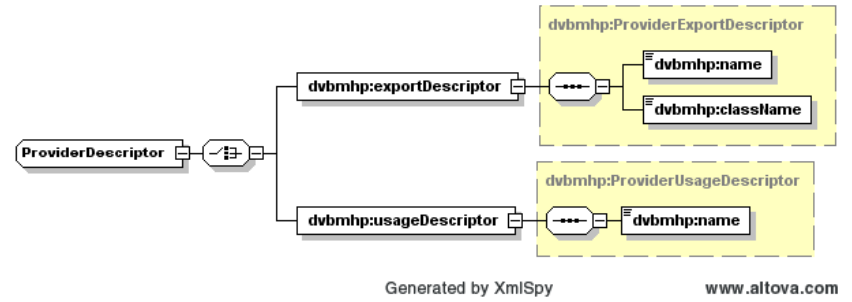
The information carried in these complex types is the same as that carried in the provider_export_descriptor and provider_usage_descriptor defined in MHP [1], clause 10.15, "Signalling for providers".

```
<xsd:complexType name="ProviderDescriptor">
    <xsd:choice>
        <xsd:element name="exportDescriptor" type="dvbmhp:ProviderExportDescriptor"/>
        <xsd:element name="usageDescriptor" type="dvbmhp:ProviderUsageDescriptor"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="ProviderExportDescriptor">
    <xsd:sequence>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="className" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ProviderUsageDescriptor">
    <xsd:sequence>
        <xsd:element name="name" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
```


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



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

Annex AS (informative): IPTV Use-cases

AS.1 Simple use-case

Table 90 breaks a simple use-case down into a series of steps. For each step, the means by which GEM supports that step is described and one or more references to the appropriate clauses provided.

This use case is for a GEM terminal supporting the privileged application option.

Table 90: Simple Use-case

Step	How supported	References
Box is turned on and boots operator application	Signalling identifies operator applications. Loaded and started by existing carousel or HTTP mechanisms.	MHP [1], clause 10.16.2, "XAIT"
Application asks for list of IPTV services via SD&S	SIManager.filterServices(DeliverySystemTypeFilter) ServiceDetails.getName	11.6.5.4, "Extensions"
Application presents list of services to user and asks to pick one	Existing GEM APIs	-
Application shows a preview of the content available on that service	Manager.createPlayer(new MediaLocator("rtsp:..."))	11.11.4.4, "Content referencing for IPTV"
Confirm purchase	Via https connection to web server using existing java.net	11.6.4, "Conditional access API"
Selects service	Service selection API	11.11.4.4, "Content referencing for IPTV"
Application removes its UI	Existing GEM (org.havi.ui or java.awt)	-
Content presented in TV mode	Existing GEM	-
Any service bound applications are now presented	Existing GEM	-
User uses pause / continue / fast-forwards / fast-rewind etc.	Either Player.setRate or TCP connection to server using java.net	11.3.1.5, "java.net", 11.4.2.10, "Additional and modified semantics for IPTV"
In parallel, operator application(s) listens for modal (menu/home) keys	org.dvb.event with new VK_ codes defined in the present document	Table 86 "Minimum set of input events"
Operator application displays its UI or changes channel	Existing GEM (java.awt or service selection API)	-
Operator application stops content being presented	Existing GEM service selection API	-

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

Annex AT (normative): Application Management API

Package org.dvb.application.privileged

Description

Provides application management facilities to privileged applications.

Class Summary	
Interfaces	
ApplicationStorageHandler	Enables privileged applications to make decisions for some methods in the org.dvb.application.storage package.
Classes	
ApplicationStorageManager	Provides application management features and capabilities only appropriate for privileged applications.

org.dvb.application.privileged ApplicationStorageHandler

Declaration

```
public interface ApplicationStorageHandler
```

Description

Enables privileged applications to make decisions for some methods in the org.dvb.application.storage package. The methods in this interface correspond to the methods in ApplicationStorageController that throw a UserRejectedInstall-Exception. If an application storage handler does not permit a request then it should return false and the original method shall fail with a UserRejectedInstallException.

Implementations of the methods in this interface should complete quickly and not perform time-consuming operations such as blocking I/O calls.

Methods

removeAppRequested(AppAttributes)

```
public boolean removeAppRequested(org.dvb.application.AppAttributes app)
```

Called as part of the execution of ApplicationStorageController.remove(AppID).

Parameters:

app - the application to be removed.

Returns:

true if the remove request is permitted, otherwise false.

removeAppRequested(AppAttributes[])

```
public boolean removeAppRequested(org.dvb.application.AppAttributes[] app)
```

Called as part of the execution of ApplicationStorageController.remove(AppID[]).

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

Parameters:

app - the applications to be removed.

Returns:

true if the remove request is permitted, otherwise false.

removeServiceRequested(ApplicationStorageController)

```
public boolean removeServiceRequested(org.dvb.application.storage.ApplicationStorageController
service)
```

Called as part of the execution of ApplicationStorageController.removeService.

Parameters:

service - the service for which removeService was called.

Returns:

true if the remove request is permitted, otherwise false.

storeRequested(ApplicationStorageController, AppAttributes[], boolean[], String[][])

```
public boolean storeRequested(org.dvb.application.storage.ApplicationStorageController service,
org.dvb.application.AppAttributes[] app, boolean[] autoStart, java.lang.String[][] args)
```

Called as part of the execution of ApplicationStorageController.store(AppProxy[],...).

Parameters:

service - the service for which store was called.

app - the applications to be installed.

autoStart - the array of booleans as passed in the call to the store method.

args - an copy of the corresponding parameter passed to the store method where each entry is .equals to the corresponding entry passed in the call to the store method.

Returns:

true if the storage request is permitted, otherwise false.

storeRequested(ApplicationStorageController, AppAttributes, boolean, String[])

```
public boolean storeRequested(org.dvb.application.storage.ApplicationStorageController service,
org.dvb.application.AppAttributes app, boolean autoStart, java.lang.String[] args)
```

Called as part of the execution of ApplicationStorageController.store(AppProxy, etc.).

Parameters:

service - the service for which store was called.

app - the application to be installed.

autoStart - the parameter as passed in the call to the store method.

args - an array of strings where each entry is .equals to the corresponding entry passed in the call to the store method.

Returns:

true if the storage request is permitted, otherwise false.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 827 of 861

storeRequested(Locator, ApplicationStorageController, AppAttributes[], boolean[], String[][])

```
public boolean storeRequested(org.davic.net.Locator fromService,
org.dvb.application.storage.ApplicationStorageController toService,
org.dvb.application.AppAttributes[] app, boolean[] autoStart, java.lang.String[][] args)
```

Called as part of the execution of ApplicationStorageController.store(Locator,AppID[], etc.).

Parameters:

`fromService` - the service containing the applications to be stored.

`toService` - the service into which the applications are to be stored.

`app` - the applications to be installed.

`autoStart` - the array of booleans as passed in the call to the store method.

`args` - an copy of the corresponding parameter passed to the store method where each entry is .equals to the corresponding entry passed in the call to the store method.

Returns:

true if the storage request is permitted, otherwise false.

org.dvb.application.privileged ApplicationStorageManager

Declaration

```
public class ApplicationStorageManager extends org.ocap.application.AppManagerProxy
```

```
java.lang.Object
  |
  +--org.ocap.application.AppManagerProxy
        |
        +--org.dvb.application.privileged.ApplicationStorageManager
```

Description

Provides application management features and capabilities only appropriate for privileged applications. On systems where this class is supported, org.ocap.application.AppManagerProxy.getInstance shall return an instance of ApplicationStorageManager.

Constructors

ApplicationStorageManager()

```
protected ApplicationStorageManager()
```

This constructor is not to be used by applications.

Methods

registerUnboundApp(Element)

```
public void registerUnboundApp(org.dvb.xml.jdom.Element element)
```

Registers new unbound application entries.

Parameters:

`element` - an ApplicationList element containing the new or updated XAIT information.

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

Throws:

`java.lang.SecurityException` - if the caller does not have MonitorAppPermission("registrar").

`java.lang.IllegalArgumentException` - if the element is not a root element whose value is "Application-List".

setApplicationStorageHandler(ApplicationStorageHandler)

```
public void setApplicationStorageHandler(org.dvb.application.privileged.ApplicationStorageHandler handler)
```

Registers an ApplicationStorageHandler to become part of decisions about permitting the storing and removing of applications and services using the facilities provided by the org.dvb.application.storage package. Only one handler can be registered at one time hence multiple calls to this method shall replace the handler set in the previous call. No handler being set shall be equivalent to a handler which always permits requests to succeed.

Parameters:

`handler` - either an ApplicationStorageHandler or null to remove the current handler if one is set.

Throws:

`java.lang.SecurityException` - if the calling application does not have MonitorAppPermission("registrar").

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

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

Package

org.dvb.locator.ip

Description

Contains transport dependent locators for access to IP services.

Class Summary	
Classes	
MulticastLocator	Represents a reference to an IP service accessed by an IGMP join request.
RTSPLocator	Represents a reference to an IP service accessed by RTSP.
UnicastLocator	Represents a reference to an IP service delivered by unicast UDP where the delivery of the service is not initiated by the implementation.

org.dvb.locator.ip

MulticastLocator

Declaration

```
public class MulticastLocator extends org.davic.net.Locator
```

```
java.lang.Object
  |
  +--org.davic.net.Locator
        |
        +--org.dvb.locator.ip.MulticastLocator
```

All Implemented Interfaces:

```
javax.tv.locator.Locator
```

Description

Represents a reference to an IP service accessed by an IGMP join request.

Constructors

MulticastLocator(InetAddress, int)

```
public MulticastLocator(java.net.InetAddress address, int port)
throws MalformedURLException
```

Creates a `MulticastLocator` from the specified URL.

Parameters:

`address` - the multicast address to join.

`port` - the port to join.

Throws:

`java.net.MalformedURLException` - if the address is not a multicast address.

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

MulticastLocator(InetAddress, int, int)

```
public MulticastLocator(java.net.InetAddress address, int port, int serviceID)
throws MalformedURLException, IllegalArgumentException
```

Creates a `MulticastLocator` from the specified URL.

Parameters:

`address` - the multicast address to join.

`port` - the port to join.

`serviceID` - a DVB-SI serviceID or MPEG program number to identify one service in a multi-program transport stream.

Throws:

`java.net.MalformedURLException` - if the address is not a multicast address.

`java.lang.IllegalArgumentException` - if the service ID is less than 0 or greater than 0xffff.

Methods

getServiceID()

```
public int getServiceID()
```

Returns the serviceID specified when this `MulticastLocator` was constructed.

Returns:

the serviceID specified when this `MulticastLocator` or -1 if no serviceID was specified.

org.dvb.locator.ip

RTSPLocator

Declaration

```
public class RTSPLocator extends org.davic.net.Locator
```

```
java.lang.Object
  |
  +--org.davic.net.Locator
        |
        +--org.dvb.locator.ip.RTSPLocator
```

All Implemented Interfaces:

```
javax.tv.locator.Locator
```

Description

Represents a reference to an IP service accessed by RTSP.

Constructors

RTSPLocator(String)

```
public RTSPLocator(java.lang.String url)
throws MalformedURLException
```

Creates a `RTSPLocator` from the specified URL.

Parameters:

`url` - an "rtsp:" url.

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

Throws:

`java.net.MalformedURLException` - if the url is not a correct "rtsp:" url.

RTSPLocator(String, int)

```
public RTSPLocator(java.lang.String url, int serviceID)
throws MalformedURLException, IllegalArgumentException
```

Creates a `RTSPLocator` from the specified URL.

Parameters:

`url` - an "rtsp:" url.

`serviceID` - a DVB-SI serviceID or MPEG program number to identify one service in a multi-program transport stream.

Throws:

`java.net.MalformedURLException` - if the url is not a correct "rtsp:" url.

`java.lang.IllegalArgumentException` - if the service ID is less than 0 or greater than 0xffff.

Methods

getServiceID()

```
public int getServiceID()
```

Returns the serviceID specified when this `RTSPLocator` was constructed.

Returns:

the serviceID specified when this `RTSPLocator` or -1 if no serviceID was specified.

org.dvb.locator.ip

UnicastLocator

Declaration

```
public class UnicastLocator extends org.davic.net.Locator
```

```
java.lang.Object
  |
  +--org.davic.net.Locator
        |
        +--org.dvb.locator.ip.UnicastLocator
```

All Implemented Interfaces:

```
javax.tv.locator.Locator
```

Description

Represents a reference to an IP service delivered by unicast UDP where the delivery of the service is not initiated by the implementation. The only requirement on implementations when locators of this type are used is to listen on the specified port number for content to arrive.

The intended use case for this locator is as follows:

- the application picks a local port number;
- the application uses this locator to tell the implementation to listen for an IP service delivered on that local port number;

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 832 of 861

- the application obtains the IP address of the device on which it's running - e.g. via java.net.NetworkInterface;
- the application passes that port number and the IP address to a VoD server as part of a request for some content;
- the VoD server delivers the requested content to the specified IP address and port;
- the implementation detects the arrival of the content on the port and starts processing and presenting it.

The external form of this locator is not defined by GEM.

Constructors

UnicastLocator(int)

```
public UnicastLocator(int port)
```

Creates a `UnicastLocator` from the specified local port number.

Parameters:

`port` - the local port number on which the content will be delivered.

UnicastLocator(int, int)

```
public UnicastLocator(int port, int serviceID)
throws MalformedURLException, IllegalArgumentException
```

Creates a `UnicastLocator` from the specified local port number and service ID.

Parameters:

`port` - the local port number on which the content will be delivered.

`serviceID` - a DVB-SI serviceID or MPEG program number to identify one service in a multi-program transport stream.

Throws:

`java.net.MalformedURLException` - if the port number is not valid.

`java.lang.IllegalArgumentException` - if the service ID is less than 0 or greater than 0xffff.

Methods

getPort()

```
public int getPort()
```

Returns the port specified when this `UnicastLocator` was constructed.

Returns:

the port specified when this `UnicastLocator` or -1 if no port was specified.

getServiceID()

```
public int getServiceID()
```

Returns the serviceID specified when this `UnicastLocator` was constructed.

Returns:

the serviceID specified when this `UnicastLocator` or -1 if no serviceID was specified.

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

AU.2. OTT content referencing API

Package org.dvb.locator.ott

Description

Contains transport dependent locators for access to OTT services.

Class Summary	
Classes	
HTTPLocator	Represents a reference to an IP service accessed by HTTP.

org.dvb.locator.ott HTTPLocator

Declaration

```
public class HTTPLocator extends org.davic.net.Locator

java.lang.Object
  |
  +--org.davic.net.Locator
        |
        +--org.dvb.locator.ott.HTTPLocator
```

All Implemented Interfaces:

```
javax.tv.locator.Locator
```

Description

Represents a reference to an IP service accessed by HTTP.

Constructors

HTTPLocator(String)

```
public HTTPLocator(java.lang.String url)
throws MalformedURLException
```

Creates a `HTTPLocator` from the specified URL.

Parameters:

`url` - an "HTTP:" url.

Throws:

`java.net.MalformedURLException` - if the url is not a correct "HTTP:" url.

HTTPLocator(String, int)

```
public HTTPLocator(java.lang.String url, int serviceID)
throws MalformedURLException, IllegalArgumentException
```

Creates a `HTTPLocator` from the specified URL.

Parameters:

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

`url` - an "HTTP:" url.

`serviceID` - a DVB-SI serviceID or MPEG program number to identify one service in a multi-program transport stream.

Throws:

`java.net.MalformedURLException` - if the url is not a correct "HTTP:" url.

`java.lang.IllegalArgumentException` - if the service ID is less than 0 or greater than 0xffff.

Methods

getServiceID()

```
public int getServiceID()
```

Returns the serviceID specified when this `HTTPLocator` was constructed.

Returns:

the serviceID specified when this `HTTPLocator` or -1 if no serviceID was specified.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 835 of 861

Annex AV (normative): Extended service list API

Package

org.dvb.service

Description

Contains features which are logically extensions to those provided by the javax.tv.service package.

Class Summary	
Interfaces	
`TransportDependentService`	Extensions to `Service` specific to transport dependent services.
`TransportIndependentService`	Extensions to `Service` specific to transport independent services.
`UnboundApplicationService`	Represents a service used to contain unbound applications.
Classes	
`DvbSIManager`	Extensions to `SIManager` for environments where a single device may support content delivery mechanisms with different types of delivery system.
`IPServiceInformationType`	IP extensions to the Java TV service information type

org.dvb.service

DvbSIManager

Declaration

```
public abstract class DvbSIManager extends javax.tv.service.SIManager implements
org.dvb.service.sds.SDSRecordAccess, org.dvb.tvanytime.javatv.CRIDAccess

java.lang.Object
  |
  +--javax.tv.service.SIManager
        |
        +--org.dvb.service.DvbSIManager
```

All Implemented Interfaces:

```
org.dvb.tvanytime.javatv.CRIDAccess, org.dvb.service.sds.SDSRecordAccess
```

Description

Extensions to `SIManager` for environments where a single device may support content delivery mechanisms with different types of delivery system.

Constructors

DvbSIManager()

```
protected DvbSIManager()
```

Constructs a DvbSIManager object

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

Methods

getTransports(DeliverySystemType)

```
public abstract javax.tv.service.transport.Transport[]
getTransports(javax.tv.service.navigation.DeliverySystemType type)
```

Reports the various content delivery mechanisms currently available on this platform which are accessed via delivery systems of the specified type. If no such transports are available then an array of length zero is returned.

Parameters:

`type` - a delivery system type.

Returns:

an array of Transport objects representing the content delivery mechanisms currently available on this platform accessed via delivery systems of the specified type.

org.dvb.service

IPServiceInformationType

Declaration

```
public class IPServiceInformationType extends javax.tv.service.ServiceInformationType
```

```
java.lang.Object
  |
  +--javax.tv.service.ServiceInformationType
        |
        +--org.dvb.service.IPServiceInformationType
```

Description

IP extensions to the Java TV service information type.

Fields

BCG

```
public static final org.dvb.service.IPServiceInformationType BCG
```

Service information type for DVB Broadband Content Guide.

SDandS

```
public static final org.dvb.service.IPServiceInformationType SDandS
```

Service information type for DVB SD&S.

Constructors

IPServiceInformationType(String)

```
protected IPServiceInformationType(java.lang.String name)
```

Creates a service information type object.

Parameters:

`name` - the string name of this type.

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

org.dvb.service TransportDependentService

Declaration

```
public interface TransportDependentService extends javax.tv.service.Service
```

All Superinterfaces:

```
javax.tv.service.Service
```

Description

Extensions to `Service` specific to transport dependent services.

Methods

getDeliverySystemType()

```
public javax.tv.service.navigation.DeliverySystemType getDeliverySystemType()
```

Return the delivery system type by which the service is accessed.

Returns:

a `DeliverySystemType`

retrieveDetails(SIRequestor)

```
public javax.tv.service.SIRequest retrieveDetails(javax.tv.service.SIRequestor requestor)
```

This method retrieves additional information about the Service. This information is retrieved from the service information in the network. This method returns data asynchronously.

Overrides:

`retrieveDetails` in interface `Service`

Parameters:

`requestor` - The SIRequestor to be notified when this retrieval operation completes.

Returns:

An SIRequest object identifying this asynchronous retrieval request.

org.dvb.service TransportIndependentService

Declaration

```
public interface TransportIndependentService extends javax.tv.service.Service
```

All Superinterfaces:

```
javax.tv.service.Service
```

Description

Extensions to `Service` specific to transport independent services.

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

Methods

transformService()

```
public org.dvb.service.TransportDependentService[] transformService()
```

Returns all the `TransportDependentService`s to which this Service can be resolved.

Returns:

an array of `TransportDependentService` instances.

transformService(DeliverySystemType)

```
public org.dvb.service.TransportDependentService[]
transformService(javax.tv.service.navigation.DeliverySystemType type)
```

Return those `TransportIndependentService`s to which this Service can be resolved which are accessed via the specified `DeliverySystemType`. If there are no such Services then an array of length zero is returned.

Parameters:

`type` - a `DeliverySystemType` instance.

Returns:

an array of TransportDependentService instances.

org.dvb.service

UnboundApplicationService

Declaration

```
public interface UnboundApplicationService extends org.ocap.service.AbstractService,
org.dvb.application.storage.ApplicationStorageController
```

All Superinterfaces:

```
org.ocap.service.AbstractService, org.dvb.application.storage.ApplicationStorageController,
javax.tv.service.Service
```

Description

Represents a service used to contain unbound applications.

Since:

MHP 1.2.

Methods

getPriority()

```
public int getPriority()
```

Get the resource priority for this service between 0 and 255.

Returns:

the resource priority for this service.

setPriority(int)

```
public void setPriority(int priority)
```

Set the resource priority for this service between 0 and 255.

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

Parameters:

`priority` - the resource priority for this service.

Throws:

`java.lang.IllegalArgumentException` - if the priority is outside this specified range.

Package org.dvb.service.navigation

Description

This package contains features which are logically extensions to those provided by the javax.tv.service.navigation package. These are mainly additional ways of filtering the service list. These are focused on more complex situations than found at the time JavaTV was developed, e.g. devices with multiple delivery system types or multiple service providers or multiple video and audio codecs.

Class Summary	
Classes	
`CompatibleServiceFilter`	CompatibleServiceFilter represents a ServiceFilter based on the capabilities of the receiver.
`DeliverySystemTypeFilter`	DeliverySystemTypeFilter represents a ServiceFilter based on a particular DeliverySystemType.
`IPDeliverySystemType`	IP extensions to the Java TV delivery system type
`ServiceProviderFilter`	ServiceProviderFilter represents a ServiceFilter based on a particular ServiceProviderInformation.

org.dvb.service.navigation CompatibleServiceFilter

Declaration

```
public final class CompatibleServiceFilter extends javax.tv.service.navigation.ServiceFilter

java.lang.Object
  |
  +--javax.tv.service.navigation.ServiceFilter
        |
        +--org.dvb.service.navigation.CompatibleServiceFilter
```

Description

CompatibleServiceFilter represents a ServiceFilter based on the capabilities of the receiver. A ServiceList resulting from this filter will include only services which are compatible with this receiver.

- If the service is scrambled by a conditional access system or equivalent, the service is only compatible if the receiver would be able to descramble the service given appropriate rights or entitlements.
- If the service includes one or more streams of type video, it is only compatible if the receiver can decode at least one stream.
- If the service includes one or more streams of type audio, it is only compatible if the receiver can decode at least one stream.

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

Constructors

CompatibleServiceFilter()

```
public CompatibleServiceFilter()
```

Constructs the filter.

Methods

accept(Service)

```
public boolean accept(javax.tv.service.Service service)
```

Tests if the given service passes the filter.

Overrides:

`accept` in class `ServiceFilter`

Parameters:

`service` - An individual `Service` to be evaluated against the filtering algorithm.

Returns:

true if service is compatible with this receiver; false otherwise.

org.dvb.service.navigation DeliverySystemTypeFilter

Declaration

```
public final class DeliverySystemTypeFilter extends javax.tv.service.navigation.ServiceFilter
```

```
java.lang.Object
  |
  +--javax.tv.service.navigation.ServiceFilter
        |
        +--org.dvb.service.navigation.DeliverySystemTypeFilter
```

Description

DeliverySystemTypeFilter represents a ServiceFilter based on a particular DeliverySystemType. A ServiceList resulting from this filter will include only Service objects that can be accessed via delivery systems of the specified `DeliverySystemType`.

Constructors

DeliverySystemTypeFilter(DeliverySystemType)

```
public DeliverySystemTypeFilter(javax.tv.service.navigation.DeliverySystemType type)
```

Constructs the filter based on a particular DeliverySystemType.

Parameters:

`type` - A `DeliverySystemType` object indicating the delivery system type for the services to be included in a resulting service list.

Methods

accept(Service)

```
public boolean accept(javax.tv.service.Service service)
```


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

Tests if the given service passes the filter.

Overrides:

accept in class ServiceFilter

Parameters:

service - An individual Service to be evaluated against the filtering algorithm.

Returns:

true if service can be accessed via a delivery system of the specified type; false otherwise.

getFilterValue()

```
public javax.tv.service.navigation.DeliverySystemType getFilterValue()
```

Reports the DeliverySystemType used to create this filter.

Returns:

The DeliverySystemType used to create this filter.

org.dvb.service.navigation IPDeliverySystemType

Declaration

```
public class IPDeliverySystemType extends javax.tv.service.navigation.DeliverySystemType
```

```
java.lang.Object
  |
  +--javax.tv.service.navigation.DeliverySystemType
        |
        +--org.dvb.service.navigation.IPDeliverySystemType
```

Description

IP extensions to the Java TV delivery system type.

Fields

IP

```
public static final org.dvb.service.navigation.IPDeliverySystemType IP
```

Delivery system type for IP.

Constructors

IPDeliverySystemType(String)

```
protected IPDeliverySystemType(java.lang.String name)
```

Creates a delivery system type object.

Parameters:

name - the string name of this type.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 842 of 861

org.dvb.service.navigation

ServiceProviderFilter

Declaration

```
public final class ServiceProviderFilter extends javax.tv.service.navigation.ServiceFilter
```

```
java.lang.Object
  |
  +--javax.tv.service.navigation.ServiceFilter
        |
        +--org.dvb.service.navigation.ServiceProviderFilter
```

Description

ServiceProviderFilter represents a ServiceFilter based on a particular ServiceProviderInformation. A ServiceList resulting from this filter will include only Service objects that can be provided by the specified service provider.

Constructors

ServiceProviderFilter(ServiceProviderInformation)

```
public ServiceProviderFilter(javax.tv.service.navigation.ServiceProviderInformation provider)
```

Constructs the filter based on a particular ServiceProviderInformation.

Parameters:

`provider` - A `ServiceProviderInformation` object indicating the service provider for the services to be included in a resulting service list.

Methods

accept(Service)

```
public boolean accept(javax.tv.service.Service service)
```

Tests if the given service passes the filter.

Overrides:

`accept` in class `ServiceFilter`

Parameters:

`service` - An individual `Service` to be evaluated against the filtering algorithm.

Returns:

true if service can be provided by the specified service provider; false otherwise.

getFilterValue()

```
public javax.tv.service.navigation.ServiceProviderInformation getFilterValue()
```

Reports the `ServiceProviderInformation` used to create this filter.

Returns:

The `ServiceProviderInformation` used to create this filter.

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

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

Package

org.dvb.service.sds

Description

Extensions to the JavaTV SI API giving access to information signalled according to DVB's service discovery and selection (SD&S) specification.

Class Summary	
Interfaces	
SDSRecordAccess	This interface provides access to Service Discovery and Selection Service Provider and Service Offering Records.
SDSRecordChangeListener	This interface is implemented by applications wishing to receive notification of changes to SD&S records
Classes	
SDSRecordChangeEvent	A SDSRecordEvent notifies an SDSRecordListener of changes detected in SD&S records.

org.dvb.service.sds

SDSRecordAccess

Declaration

```
public interface SDSRecordAccess
```

All Known Implementing Classes:

```
org.dvb.service.DvbSIManager
```

Description

This interface provides access to Service Discovery and Selection Service Provider and Service Offering Records. It is intended to be implemented by an instance of org.dvb.service.SIManager.

Methods

addSDSRecordListener(SDSRecordChangeListener)

```
public void addSDSRecordListener(org.dvb.service.sds.SDSRecordChangeListener listener)
```

Register a listener for events based on changes to SD&S records.

Parameters:

`listener` - the SDS Record Listener to be registered.

getServiceOfferingRecords(String)

```
public org.dvb.xml.jdom.Element[] getServiceOfferingRecords(java.lang.String domainName)
```

Provides access to the SD&S Service Offering records for a particular Service Provider.

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

Parameters:

domainName - Domain name of the service provider for which offerings are requested.

Returns:

an array of SD&S Service Offering Records, including BCG records.

getServiceProviderRecords()

```
public org.dvb.xml.jdom.Element[] getServiceProviderRecords()
```

Provides access to the SD&S Service Provider records.

Returns:

an array of SD&S Service Provider Records.

removeSDSRecordListener(SDSRecordChangeListener)

```
public void removeSDSRecordListener(org.dvb.service.sds.SDSRecordChangeListener listener)
```

Removes a listener for events based on changes to SD&S records.

Parameters:

listener - the SDS Record Listener to be removed.

org.dvb.service.sds

SDSRecordChangeEvent

Declaration

```
public abstract class SDSRecordChangeEvent extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--org.dvb.service.sds.SDSRecordChangeEvent
```

All Implemented Interfaces:

```
java.io.Serializable
```

Description

A SDSRecordEvent notifies an SDSRecordListener of changes detected in SD&S records. Specifically, this event signals the addition, removal, or modification of an SD&S record.

Constructors

SDSRecordChangeEvent(DvbSIManager, SIChangeType, Element)

```
public SDSRecordChangeEvent(org.dvb.service.DvbSIManager siManager,
javax.tv.service.SIChangeType type, org.dvb.xml.jdom.Element sdsRecord)
```

Constructs an SDSRecordChangeEvent object.

Parameters:

siManager - the SIManager on which the Event initially occurred.

type - the type of change that occured.

sdsRecord - the SD&S record in which the change occurred.

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

Methods

getChangeType()

```
public javax.tv.service.SIChangeType getChangeType()
```

Returns the type of change that occurred.

Returns:

SIChangeType the type of change that occurred.

getSDSRecord()

```
public org.dvb.xml.jdom.Element getSDSRecord()
```

Returns the SD&S record in which the change occured.

Returns:

Document XML document of the SD&S record in which the change occurred.

org.dvb.service.sds

SDSRecordChangeListener

Declaration

```
public interface SDSRecordChangeListener extends java.util.EventListener
```

All Superinterfaces:

```
java.util.EventListener
```

Description

This interface is implemented by applications wishing to receive notification of changes to SD&S records.

Methods

notifyChange(SDSRecordChangeEvent)

```
public void notifyChange(org.dvb.service.sds.SDSRecordChangeEvent event)
```

Notification of a change in either an SDS Service Provider or Service Offering record.

Parameters:

`event` - the event that caused the change.

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

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

In the present document, `DvbLocator` references are to be replaced with references to `org.davic.net.Locator` or an implementation-defined subclass.

Package org.dvb.tvanytime.metadata.ip

Description

Provides access to Broadband Content Guides delivered via IP.

Class Summary	
Classes	
`BCGDiscovery`	Class providing access to Broadband Content Guides whose location is signalled by IP-based Service Discovery & Selection information.
`HTTPDatabase`	Class providing access to a Broadband Content Guide delivered using the unicast delivery method specified in TS 102 034 [80] (V1.2.1).
`MulticastDatabase`	Class providing access to a Broadband Content Guide delivered using the multicast delivery method specified in TS 102 034 [80] (V1.2.1).

org.dvb.tvanytime.metadata.ip BCGDiscovery

Declaration

```
public abstract class BCGDiscovery

java.lang.Object
  |
  +--org.dvb.tvanytime.metadata.ip.BCGDiscovery
```

Description

Class providing access to Broadband Content Guides whose location is signalled by IP-based Service Discovery & Selection information.

Constructors

BCGDiscovery()

```
public BCGDiscovery()
```

This constructor is provided for implementations and should not be used by GEM applications.

Methods

getAvailableDatabases(CRID)

```
public static org.dvb.tvanytime.metadata.Database[]
getAvailableDatabases(org.dvb.tvanytime.resolution.CRID crid)
throws DatabaseException
```

Get an array of Databases that can provide metadata for the specified CRID Authority.

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

Parameters:

`crid` - the authority field from the supplied CRID is used to decide which metadata servers to return. The CRID authority should be matched against Metadata Pointer Descriptors in Resolution Provider Notification Tables delivered by IP-based Service Discovery & Selection information or in a DVB service.

Returns:

an array of type `org.dvb.tvanytime.metadata.Database` . Each object in the returned array will be an instance of either `MulticastDatabase` , `HTTPDatabase` or `org.dvb.tvanytime.metadata.IPDatabase`.

Throws:

`org.dvb.tvanytime.metadata.DatabaseException` - if no metadata services can be found for the specified scope.

getAvailableDatabases(Locator)

```
public static org.dvb.tvanytime.metadata.Database[] getAvailableDatabases(javax.tv.locator.Locator
locator)
throws DatabaseException
```

Get an array of Databases that can provide metadata for the specified service.

Parameters:

`locator` - a locator representing a service. This should be matched against entries in IP-based Service Discovery & Selection information to find Broadband Content Guides identified by ServiceDescriptionLocation and Services-DescriptionLocation fields.

Returns:

an array of type `org.dvb.tvanytime.metadata.Database` . Each object in the returned array will be an instance of either `MulticastDatabase` , `HTTPDatabase` or `org.dvb.tvanytime.metadata.IPDatabase`.

Throws:

`org.dvb.tvanytime.metadata.DatabaseException` - if no metadata services can be found for the specified scope.

getAvailableDatabases(String)

```
public static org.dvb.tvanytime.metadata.Database[] getAvailableDatabases(java.lang.String
serviceProviderDomainName)
throws DatabaseException
```

Get an array of Databases that can provide metadata for the specified service provider.

Parameters:

`serviceProviderDomainName` - a String representing the domain name of a service provider. This should be matched against the TargetProvider fields in Broadband Content Guide Discovery Records delivered by IP-based Service Discovery & Selection information.

Returns:

an array of type `org.dvb.tvanytime.metadata.Database` . Each object in the returned array will be an instance of either `MulticastDatabase` , `HTTPDatabase` or `org.dvb.tvanytime.metadata.IPDatabase`.

Throws:

`org.dvb.tvanytime.metadata.DatabaseException` - if no metadata services can be found for the specified scope.

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

getAvailableDatabasesByProvider(String)

```
public static org.dvb.tvanytime.metadata.Database[] getAvailableDatabasesByProvider(java.lang.String
BCGProviderName)
throws DatabaseException
```

Get an array of Databases from the specified BCG provider.

Parameters:

`BCGProviderName` - a String representing the domain name of the BCG provider. This should be matched against the BCGProviderName in Broadband Content Guide Discovery Records delivered by IP-based Service Discovery & Selection information.

Returns:

an array of type `org.dvb.tvanytime.metadata.Database` . Each object in the returned array will be an instance of either `MulticastDatabase` , `HTTPDatabase` or `org.dvb.tvanytime.metadata.IPDatabase`.

Throws:

`org.dvb.tvanytime.metadata.DatabaseException` - if no metadata services can be found for the specified scope.

getDatabase()

```
public static org.dvb.tvanytime.metadata.Database getDatabase()
throws DatabaseException
```

Get an instance of a Broadband Content Guide where the platform chooses the host.

Returns:

a Database.

Throws:

`org.dvb.tvanytime.metadata.DatabaseException` - if a metadata service cannot be found.

org.dvb.tvanytime.metadata.ip

HTTPDatabase

Declaration

```
public abstract class HTTPDatabase implements org.dvb.tvanytime.metadata.Database
```

```
java.lang.Object
  |
  +--org.dvb.tvanytime.metadata.ip.HTTPDatabase
```

All Implemented Interfaces:

`org.dvb.tvanytime.metadata.Database`

Description

Class providing access to a Broadband Content Guide delivered using the unicast delivery method specified in TS 102 034 [80] (V1.2.1).

Constructors

HTTPDatabase()

```
public HTTPDatabase()
```

This constructor is provided for implementations and should not be used by GEM applications.

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

Methods

getURL()

```
public org.davic.net.Locator getURL()
```

Get the URL of the host providing this HTTPDatabase.

Returns:

the URL.

newInstance(String)

```
public static org.dvb.tvanytime.metadata.ip.HTTPDatabase newInstance(java.lang.String path)
throws DatabaseException
```

Get an instance of an HTTPDatabase where the caller chooses the server.

Parameters:

`path` - a URL that points to the metadata server.

Returns:

an HTTPDatabase.

Throws:

`org.dvb.tvanytime.metadata.DatabaseException` - if the platform is unable to connect to the remote server.

org.dvb.tvanytime.metadata.ip

MulticastDatabase

Declaration

```
public abstract class MulticastDatabase implements org.dvb.tvanytime.metadata.MonitoredDatabase
```

```
java.lang.Object
  |
  +--org.dvb.tvanytime.metadata.ip.MulticastDatabase
```

All Implemented Interfaces:

`org.dvb.tvanytime.metadata.Database, org.dvb.tvanytime.metadata.MonitoredDatabase`

Description

Class providing access to a Broadband Content Guide delivered using the multicast delivery method specified in TS 102 034 [80] (V1.2.1).

Constructors

MulticastDatabase()

```
public MulticastDatabase()
```

This constructor is provided for implementations and should not be used by GEM applications.

Methods

getURL()

```
public org.davic.net.Locator getURL()
```

Get the URL of the host providing this MulticastDatabase.

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

Returns:

the URL.

newInstance(String)

```
public static org.dvb.tvanytime.metadata.ip.MulticastDatabase newInstance(java.lang.String path)
throws DatabaseException
```

Get an instance of a MulticastDatabase where the caller chooses the server.

Parameters:

`path` - a URL that points to the metadata server.

Returns:

a MulticastDatabase.

Throws:

`org.dvb.tvanytime.metadata.DatabaseException` - if the platform is unable to locate the metadata service.

Ex. 1024 – TCL Indus Holdings Co., Ltd.
TCL Indus Holdings Co., Ltd. v. LGE – IPR2023-00238
Page 851 of 861

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

Package

org.dvb.tvanytime.javatv

Description

Contains TV-Anytime specific extensions to javax.tv.*.

Class Summary	
Interfaces	
CRIDAccess	This interface extends javax.tv.service.SIManager with overloaded methods that take CRID and InstanceMetadataId as parameters.
TVAProgramEvent	Extensions to ProgramEvent to enable access to extended TV-Anytime data associated with the event.

org.dvb.tvanytime.javatv

CRIDAccess

Declaration

```
public interface CRIDAccess
```

All Known Implementing Classes:

```
org.dvb.service.DvbSIManager
```

Description

This interface extends javax.tv.service.SIManager with overloaded methods that take CRID and InstanceMetadataId as parameters. It is intended to be implemented by an instance of org.dvb.service.SIManager.

Methods

retrieveProgramEvent(CRID, String, SIRequestor)

```
public javax.tv.service.SIRequest retrieveProgramEvent(org.dvb.tvanytime.resolution.CRID crid,
java.lang.String instanceMetadataId, javax.tv.service.SIRequestor requestor)
throws SecurityException
```

Retrieves all ProgramEvent associated with the CRID and optionally the instance metadata Id.

Parameters:

crid - The TV-Anytime CRID associated with the ProgramEvent(s).

instanceMetadataId - The TV-Anytime Instance Metadata Id associated with the service or null.

requestor - the requestor object to be used for the asynchronous callback.

Returns:

the SIRequest object associated with the asynchronous request.

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

Throws:

java.lang.SecurityException.

retrieveServiceDetails(CRID, String, SIRequestor)

```
public javax.tv.service.SIRequest retrieveServiceDetails(org.dvb.tvanytime.resolution.CRID crid,
java.lang.String instanceMetadataId, javax.tv.service.SIRequestor requestor)
throws SecurityException
```

Retrieves the `ServiceDetails` of all services associated with the CRID and optionally the instance metadata Id.

Parameters:

`crid` - The TV-Anytime CRID associated with the service(s).

`instanceMetadataId` - The TV-Anytime Instance Metadata Id associated with the service or null.

`requestor` - the requestor object to be used for the asynchronous callback.

Returns:

the SIRequest object associated with the asynchronous request.

Throws:

java.lang.SecurityException.

retrieveSIElement(CRID, String, SIRequestor)

```
public javax.tv.service.SIRequest retrieveSIElement(org.dvb.tvanytime.resolution.CRID crid,
java.lang.String instanceMetadataId, javax.tv.service.SIRequestor requestor)
throws SecurityException
```

Retrieves all `SIElement` that are instances of `ProgramEvent` associated with the CRID and optionally the instance metadata Id.

Parameters:

`crid` - The TV-Anytime CRID associated with the SIElement(s).

`instanceMetadataId` - The TV-Anytime Instance Metadata Id associated with the service or null.

`requestor` - the requestor object to be used for the asynchronous callback.

Returns:

the SIRequest object associated with the asynchronous request.

Throws:

java.lang.SecurityException.

org.dvb.tvanytime.javatv

TVAProgramEvent

Declaration

```
public interface TVAProgramEvent extends javax.tv.service.guide.ProgramEvent
```

All Superinterfaces:

```
javax.tv.service.guide.ProgramEvent, javax.tv.service.SIElement, javax.tv.service.SIRetrievable
```

Description

Extensions to `ProgramEvent` to enable access to extended TV-Anytime data associated with the event.

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

Methods

getCRID()

```
public org.dvb.tvanytime.resolution.CRID getCRID()
```

Returns the `CRID` associated with this Program Event.

Returns:

the CRID associated with this Program Event.

getInstanceMetadataId()

```
public java.lang.String getInstanceMetadataId()
```

Returns the `InstanceMetadataId` associated with this Program Event (if specified) or null.

Returns:

the Instance Metadata Id associated with this Program Event.

getProgramInformationElement()

```
public org.dvb.xml.jdom.Element getProgramInformationElement()
```

Returns an XML Element from the relevant Program Information table entry.

Returns:

the TV-Anytime element that points to the start of the program information entry.

getProgramLocationElement()

```
public org.dvb.xml.jdom.Element getProgramLocationElement()
```

Returns an XML Element from the relevant Program Location table entry.

Returns:

the TV-Anytime element that points to the start of the program location entry.

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

Annex AZ (normative): MHP terminal hardware API

Package

org.dvb.hardware

Description

Contains features related to the GEM terminal itself.

Class Summary	
Classes	
Terminal	Provides access to low level details of the GEM terminal.

org.dvb.hardware

Terminal

Declaration

```
public class Terminal

java.lang.Object
  |
  +--org.dvb.hardware.Terminal
```

Description

Provides access to low level details of the GEM terminal.

Constructors

Terminal()

```
protected Terminal()
```

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.

Methods

getInstance()

```
public static org.dvb.hardware.Terminal getInstance()
```

Obtain an instance of the singleton Terminal object.

Returns:

the Terminal.

getRebootTimeout()

```
public int getRebootTimeout()
```

Get the time that the system shall wait for the SystemEventListener.notifyEvent(SystemEvent) to return before continuing regardless.

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

Returns:

the reboot timeout measured in seconds.

reinitialize()

```
public void reinitialize()
```

Re-initializes the GEM software environment including re-initializing all DVB-J virtual machine or machines and re-starting all GEM applications. The method SystemEventListener.notifyEvent(SystemEvent) shall be called before the re-initialization is performed by the GEM terminal. Once the notifyEvent method returns, the process shall continue. If the notify event method does not return within an implementation dependent time then the process may continue regardless.

Throws:

`java.lang.SecurityException` - if the caller does not have MonitorAppPermission("reboot").

reset()

```
public void reset()
```

Initiates a reset of the GEM terminal equivalent to a power cycle. The method SystemEventListener.notify-Event(SystemEvent) shall be called before the reboot is performed by the GEM terminal. Once the notifyEvent method returns, the reboot process shall continue. If the notify event method does not return within an implementation dependent time then the reboot process may continue regardless.

Throws:

`java.lang.SecurityException` - if the caller does not have MonitorAppPermission("reboot").

setRebootTimeout(int)

```
public void setRebootTimeout(int seconds)
```

Set the time that the system shall wait for the SystemEventListener.notifyEvent(SystemEvent) to return before it shall continue regardless. If not set then the default is implementation dependent.

Parameters:

`seconds` - the time to wait in seconds.

Throws:

`java.lang.SecurityException` - if the caller does not have MonitorAppPermission("reboot").

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

Annex BA (normative): Content Download API

Package

org.oipf.download

Description

This package enables downloading of arbitrary content. It provides an extension of the shared DVR API [85].

Class Summary	
Classes	
LocatorDownloadSpec	Specifies a recording request in terms of a Locator to a file. The identified file is asynchronously downloaded from the network.
ApplicationDownloadSpec	Represents a content download to be performed by an application
Interfaces	
ApplicationDownloadRequest	Represents a content download to be performed by an application.
Exceptions	
ApplicationDownloadException	Thrown when an application calls methods on ApplicationDownloadRequest which it is not permitted to call.

ApplicationDownloadRequest

```
package org.oipf.download
public interface ApplicationDownloadRequest
```

Represents a content download to be performed by an application. Requests of this type are handled totally by an application. They are created in the PENDING_NO_CONFLICT_STATE. The application handling the request is responsible for downloading the content and all resulting state changes.

Requests of this type shall always be visible to the application specified when the corresponding ApplicationDownloadSpec was created regardless of recording request specific security attributes. The methods defined in this class shall always fail with a ApplicationDownloadException if called by any application other than the one specified when the corresponding ApplicationDownloadSpec was created.

getFile

```
java.io.RandomAccessFile getFile() throws ApplicationDownloadException
```

Return a file to which downloaded data can be written and from which downloaded data can be read.

Returns:

a File

Throws:

ApplicationDownloadException - if the caller is not permitted to call this method as defined above

setFailedReason

```
void setFailedReason(int reason)
```


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

Set the reason to use for the exception returned by getFailedException. The reason must be one valid for the constructor of RecordingFailedException.

Parameters:

reason - the reason to use when constructing a RecordingFailedException

setState

```
void setState(int state)
```

Set the state of the download.

Parameters:

state - the new state of the recording

Throws:

ApplicationDownloadException - if the caller is not permitted to call this method as defined above

ApplicationDownloadSpec

```
java.lang.Object
  └ org.ocap.shared.dvr.RecordingSpec
      └ org.oipf.download.ApplicationDownloadSpec
```

```
package org.oipf.download
public class ApplicationDownloadSpec extends org.ocap.shared.dvr.RecordingSpec
```

Represents a content download to be performed by an application. The application to handle the request and the source of the content to download must be specified when the request is created. No other application is permitted to handle the request. The format of the source must be co-ordinated between the application requesting the download and the application handling the download request.

ApplicationDownloadSpec

```
public ApplicationDownloadSpec(AppID app, java.lang.String source,  RecordingProperties properties)
```

Create a request for content download by application.

Parameters:

app - the application to perform the download

source - the source of the download

properties - the properties for the download

getAppID

```
public AppID getAppID()
```

Return the application ID specified.

Returns:

an application ID

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

getSource

```
public java.lang.String getSource()
```

Return the source specified.

Returns:

a String

LocatorDownloadSpec

```
java.lang.Object
  └ org.ocap.shared.dvr.RecordingSpec
      └ org.oipf.download.LocatorDownloadSpec
```

```
package org.oipf.download
public class LocatorDownloadSpec extends org.ocap.shared.dvr.RecordingSpec
```

Specifies a recording request in terms of a Locator to a file. The identified file is asynchronously downloaded from the network.

When instances of this class are passed to RecordingManager.record(..), the following additional failure mode shall apply - if the locator does not reference a file then the record method shall throw an IllegalArgumentException.

No additional failure modes are defined for RecordingRequest.reschedule.

For recording requests resulting from a recording spec of this type, downloading shall start immediately. Such recording requests will never be in a pending or a IN_PROGRESS_INCOMPLETE_STATE state.

LocatorDownloadSpec

```
public LocatorDownloadSpec(Locator source, RecordingProperties properties)
```

Constructor

Parameters:

`source` - Source of content to be downloaded

`properties` - the definition of how the recording is to be done

getSource

```
public Locator getSource()
```

Returns the source of the content to be downloaded

Returns:

the source passed into the constructor

ApplicationDownloadException

```
java.lang.Object
  └ java.lang.Throwable
```


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

```
└ java.lang.Exception
    └ org.oipf.download.ApplicationDownloadException
```

All Implemented Interfaces:

java.io.Serializable

```
package org.oipf.download
public class ApplicationDownloadException extends java.lang.Exception
```

Thrown when an application calls methods on ApplicationDownloadRequest which it is not permitted to call.

ApplicationDownloadException

```
public ApplicationDownloadException()
```

Constructs a `ApplicationDownloadException` with `null` as its error detail message.

ApplicationDownloadException

```
public ApplicationDownloadException(java.lang.String s)
```

Constructs a `ApplicationDownloadException` with the specified detail message. The error message string `s` can later be retrieved by the `Throwable.getMessage()` method of class `java.lang.Throwable`.

Parameters:

`s` - the detail message.

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

History

Document history		
V1.0	October 2010	First draft after MUG F2F
V1.1	November 2010	Second draft
V1.2	December 2010	Third draft
V1.3	January 2011	Release candidate

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