

ETSI TS 101 993 V1.2.1 (2004-08)

Technical Specification

Digital Audio Broadcasting (DAB); A Virtual Machine for DAB: DAB Java Specification

European Broadcasting Union



Union Européenne de Radio-Télévision

EBU-UER

DAB
Digital Audio Broadcasting



Reference

DTS/JTC-DAB-27

Keywords

API, audio, broadcasting, DAB, data, digital

ETSI

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

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

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

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

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

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <http://portal.etsi.org/tb/status/status.asp>

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

editor@etsi.fr

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2002.

© European Broadcasting Union 2002.

All rights reserved.

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

Contents

Intellectual Property Rights	5
Foreword	5
Introduction	5
1 Scope	6
2 References	6
3 Definitions and abbreviations	6
3.1 Definitions	6
3.2 Abbreviations	7
4 The DAB package	7
4.0 Summary	7
4.1 The communication concept	7
4.1.1 The communication between the application and the DAB package	7
4.2 Commands	8
4.3 Examples	11
4.3.1 EPG	11
4.3.2 Ticker	17
4.4 Command Types	19
4.4.1 Tuning	20
4.4.2 Searching	21
4.4.3 Scanning	22
4.4.4 Accessing service directory information	23
4.4.5 Accessing service directory information	24
4.4.6 Monitoring reception quality	25
4.4.7 Selecting an audio component	25
4.4.8 Selecting a slideshow or a dynamic label component	26
4.4.9 Selecting a broadcast website component	27
4.4.10 Selecting an object	29
4.4.11 Selecting a component stream	33
4.4.12 Operation control	34
4.4.13 Retrieving location information	35
4.5 Dependencies between the commands	35
4.6 Client registration	36
4.7 The package structure	37
5 The runtime package	43
5.0 Summary	43
5.1 The DAB Application Model	43
5.2 Control of Java applications	46
5.2.1 Packaging	46
5.2.2 Transmission as DAB-Java MOT carousel	46
5.2.2.1 MOT parameter for individual objects	46
5.2.2.2 MOT parameter for the entire carousel	47
5.2.2.2.1 The ApplicationArchive Parameter	47
5.2.2.2.2 The Class-Path Parameter	47
5.2.3 Selecting an Application	48
5.2.4 Starting an Application	48
5.2.5 Terminating an Application	48
5.2.6 Loading classes	48
5.2.7 Control of applications	48
5.2.7.1 Application context	49
5.2.7.2 Proxy	49
5.2.7.3 Example	51
5.3 Security management	53
5.4 Resource management	55
5.4.1 Model	55

5.4.2	Conflict Resolution	55
5.5	Configuration management.....	59
6	The User I/O Package	60
6.1	Signalling.....	60
6.1.1	DAB Java User Application Profile (DJUAP).....	60
6.1.2	Platform.....	61
6.1.3	Version.....	61
6.1.4	Access	61
6.1.5	Defined profiles	61
6.1.5.1	Standard Personal Java Profile (SPJP).....	61
6.1.6.2	Network enabled Personal Java Profile (NPJP).....	61
6.2	DABJava platforms.....	62
6.2.1	PersonalJava 1.1	62
6.2.1.1	Core Packages	62
6.2.1.2	DABJava profiles: specific packages.....	63
6.2.1.2.1	Standard Personal Java Profile (SPJP)	63
6.2.1.2.2	Network-enabled Personal Java Profile (NPJP).....	63
Annex A (normative):	64	
The DAB Java classes	64	
Annex B (informative): Bibliography	64	
History.....	460	

Intellectual Property Rights

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

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

Foreword

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

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

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

Introduction

This clause contains an extension of the DAB specifications to provide a Java-based software framework to develop portable applications using DAB data services.

A task force (Task Force Virtual Machine) was established inside the EUREKA 147 Consortium to discuss and to specify a Virtual Machine for executing applications. The concept of Virtual Machine is related to the requirements of providing a type of application that can be executed independently from hardware specific configuration.

Due to its strong flexibility and commercial success, Java was chosen as a base technology for solving the requirements of a portable environment and for specifying a set of API designed for the DAB environment.

A DAB extension to the Java API has been designed by the members of the Task Force for virtual Machine: such extension provides the software framework for designing, implementing and executing portable applications specifically targeted to the DAB system.

1 Scope

The present document specifies a DAB related API for Java. This API enables the download of Java programs via DAB and their control on their execution. Additionally, it provides an application level interface to the functionality of DAB.

2 References

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

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

- [1] ETSI EN 301 234 (V1.2.1 onwards): "Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) protocol".
- [2] ETSI TS 101 812: "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification".
- [3] ETSI EN 300 401: "Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers".
- [4] ETSI TS 101 756: "Digital Audio Broadcasting (DAB): Registered Tables".
- [5] ETSI TS xxx ddd: "Digital Audio Broadcasting (DAB): Data Services: DAB URL".

3 Definitions and abbreviations

3.1 Definitions

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

application controller: This entity is part of the runtime framework and is responsible for the control of a downloaded application. It acts as an intermediate between the application that initiated the download and the downloaded application.

command: transaction between a DAB client and the DAB package. It consists of a request, which is sent from the client to the package and confirmations and notifications, which are sent from the package to the client.

DAB resource: collection of hardware and software components that reside on a DAB terminal

NOTE: For example, hardware resources are audio, video, input devices, DAB receiver settings and commands; software resources are DAB Platform API access, DAB terminal API access, etc.

package: This is used throughout the text in two ways. First, it designates a Java package. Additionally, it also designates a component in DAB Java.

Comment from Thomas Hack : clarify on the wording service. sometimes it is used for the presentation of a user application (i.e. dls service, bws service, ...). The service information is derived from the DAB-service which is the container for a set of user applications. Change the wording service for the presentation of user applications.

3.2 Abbreviations

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

API	Application Programming Interface
DAB	Digital Audio Broadcasting
EPG	Electronic Programme Guide (Example Java Application)
FIG	Fast Information Group
IO	Input/Output
JDK	Java Development Kit
MOT	Multimedia Object Transfer
NPJP	Network-enabled Personal Java Profile
PAD	Programme Associated Data
SPJP	Standard Personal Java Profile
UA	User Application
VM	Virtual Machine (not only the Java Virtual Machine, but also in the sense of the whole runtime environment for DABJava)
WIRC	WorldDAB Information and Registration Centre, Wyvil Court, Wyvil Road, LONDON SW8 2TG, England, Tel: +44 171 896 90 51, Fax: +44 171 896 90 55, E-mail: worldddab-irc@worldddab.org

4 The DAB package

4.0 Summary

The DAB package offers an interface to access DAB Resources for any kind of Java application..In clause 4.1 the basic and high-level communication concept is described. The former is based on the Event-Listener pattern. The latter consists of transactions, so called commands, in which the application sends requests to the package and the package responds with confirmations and notifications. In clause 4.2 the particular commands and resulting patterns are presented. The use of these commands is exemplified in clause 4.3. For each command there is a typical interaction. This is described in clause 4.4. In clause 4.5 the dependencies between the commands are explained. The commands can only be used if the client is registered. This is shown in clause 4.6. Clause 4.7 consists of a description of the classes, which are contained in the package.

4.1 The communication concept

This clause describes the communication concepts for the use of the DAB package. The DAB package provides high-level access to the services of the Digital Audio Broadcasting (DAB) System. The package uses the Event-Listener pattern (see Bibliography, "Design Pattern, Element of Reusable Object-oriented Software by Erich Gamma") for the communication between the DAB system and the application. On top of this basic communication pattern a transaction concept is defined (e.g. to deal with ongoing events).

4.1.1 The communication between the application and the DAB package

The DAB package provides an asynchronous interface. If an application calls a method of `DABSource` (the basic interface), the result is not passed back as a return value. Instead the method initiates a new transaction and just returns. This transaction will generate events for intermediate or final results that are passed to the application using the `DABListener` interface.

This mechanism for passing back results to the application, follows the Event-Listener pattern. The package acts as a source for events that are distributed to all components which implement the `DABListener` interface and which are registered as listeners.

The following piece of code shows, how to use this in an application:

```
import dab.*;

public class MyApplication extends DABAdapter
{
    private DABClient dabClient=new DABClient();

public void start()
{
    // Configuration
    dabClient.open();
    // Registration
    dabClient.addDABListener(this);

    ...
    // Initiate request for object
    dabClient.selectObjectReq(...);

    ...
    dabClient.removeDABListener(this);
}

public void selectObjectCnf(SelectObjectCnfEvent e)
{
    // application actions on the confirm
    ...
}
}
```

The class `MyApplication` is subclassing `DABAdapter` to act as a `DABListener`. `DABAdapter` is an auxiliary class, which implements the `DABListener` with empty methods. The variable `dabClient` is used to interface to the DAB package. Before the client can be used, the connection to the receiver has to be set up using `open`. The next step is to register the application, so that it gets events. This is achieved by calling `addDABListener`. When we initiate a request like `selectObjectReq` (see table 1), the package will call our `selectObjectCnf` method for events of type `SelectObjectCnfEvent`. In the end the application calls `removeDABListener` to stop any event distribution to itself.

4.2 Commands

The basic transaction model in the DAB package is that a client issues requests and the DAB package responds with confirmations and notifications. Such a transaction is called a command. A command is initiated by a request and is finalized by a confirmation.

If notifications are sent while a command is executed, the notification informs about the progress of the transaction. Notifications are also used in situations when a DAB client requests particular information which cannot be delivered immediately or in situations where updates may occur after the first request for information has been satisfied. In the latter case notifications are delivered after the confirmation until the command is explicitly cancelled.

Table 1: Commands

Command	Request	Notification (in between)	Confirmation	Notification (following)
Pattern	Req		Cnf	
Tune	tuneReq	-	tuneCnf	-
GetEnsembleInfo	getEnsembleInfoReq	-	getEnsembleInfoCnf	-
GetServiceInfo	getServiceInfoReq	-	getServiceInfoCnf	-
GetComponentInfo	GetComponentInfoReq	-	GetComponentInfoCnf	-
SelectComponent	selectComponentReq	-	selectComponentCnf	-
SelectApplication	selectApplicationReq	-	selectApplicationCnf	-
SelectComponentStream	selectComponentStreamReq	-	selectComponentStreamCnf	-
Pattern	Req	Ntf	Cnf	
Search	searchReq	searchNtf	searchCnf	-
Scan	ScanReq	scanNtf	scanCnf	
Pattern	Req		Cnf	Ntf
SelectSI	selectSIReq	-	SelectSICnf	siNtf
SelectReceptionInfo	selectReceptionInfoReq	-	selectReceptionInfoCnf	receptionInfoNtf
SelectObject	selectObjectReq	-	selectObjectCnf	objectNtf
GetLocationInfo	getLocationInfoReq	-	getLocationInfoCnf	locationInfoNtf
OperationControl	operationControlReq	-	operationControlCnf	ServiceFollowingNtf, drcModeNtf
SystemFailure				SystemFailureNtf

To summarize this, commands are executed by sending requests, confirmations and notifications. Three different command patterns are used in the DAB package (see figure 1). All commands are listed in table 1 sorted by the pattern type (endings with Req = request, Cnf = confirmation, Ntf = notification). The patterns describe only the message sequence for one command. If commands are interleaved which means two commands running at the same time an arbitrary sequence of message types is possible.

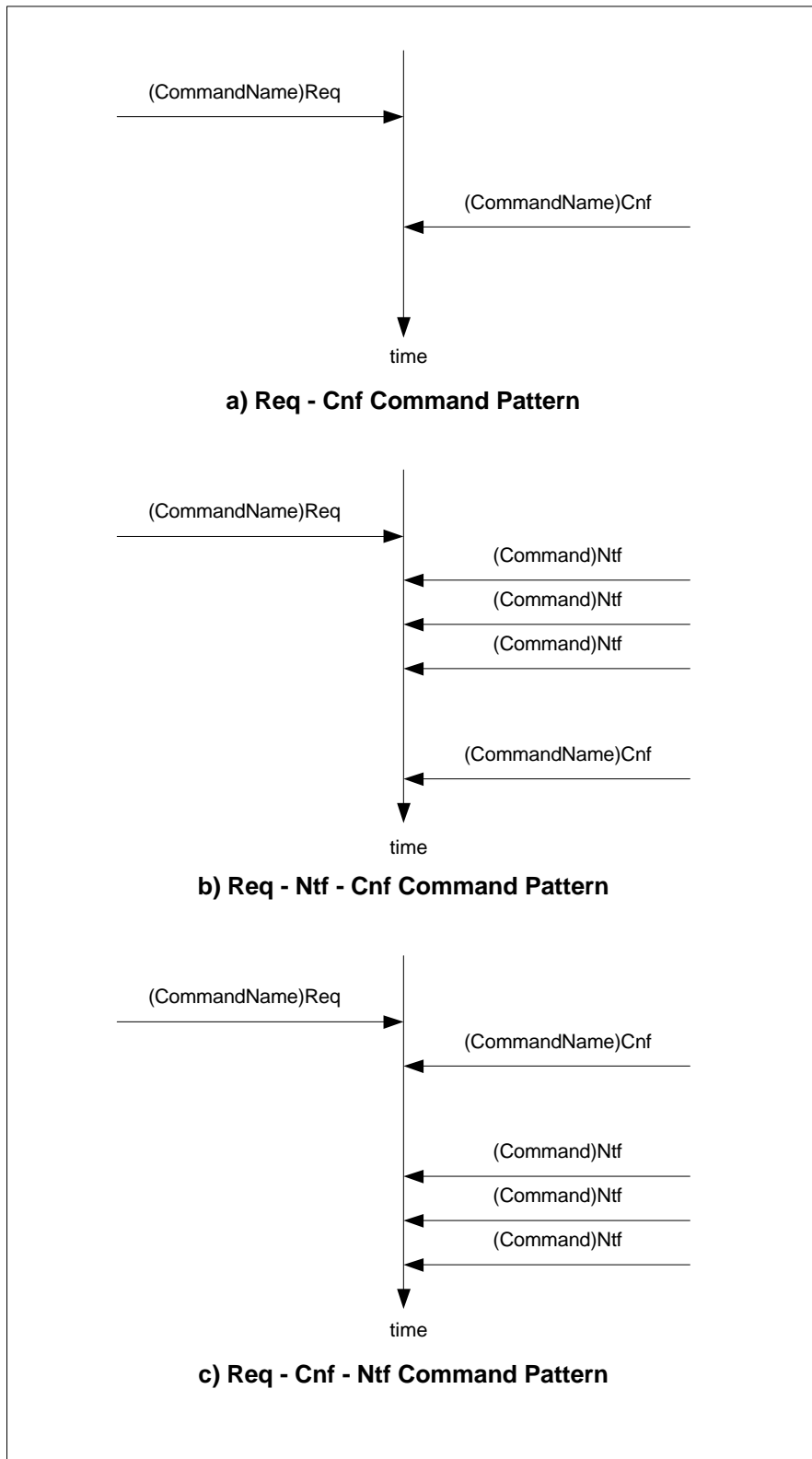


Figure 1: Command patterns

The asynchronous messaging approach fits well with the requirements of a broadcast based information system. In a broadcast system only unidirectional communication from the service provider to the client is possible. Therefore clients have to be prepared that changes might occur at anytime. A DAB ensemble is able to provide several services simultaneously. The number of services or their type can change at anytime. Also the reception conditions of a DAB receiver in a mobile environment might change very often. The DAB package keeps track on all these dynamic aspects of a broadcast system and informs a connected client by sending update notifications.

Interleaved started commands are also processed interleaved as far as the semantic of the commands allows this, e.g. it is possible to change the volume while a SelectObject command is pending. If two commands cannot be processed interleaved the one which has been sent first is also processed first.

4.3 Examples

The following examples demonstrate the use of commands in typical areas like service information access and service presentation. In the first example a simple EPG application is described. The second example shows a stock market ticker application.

4.3.1 EPG

In the following we will show how to use the DAB Java interface for an EPG application. We will focus on the main steps for initializing and controlling the DAB system. Basically, we need only two main classes: a `DABListener` (usually on the application side, in our case the class `MyEPG`) and a `DABClient` (the main entry for controlling the DAB receiver - see figure 2).

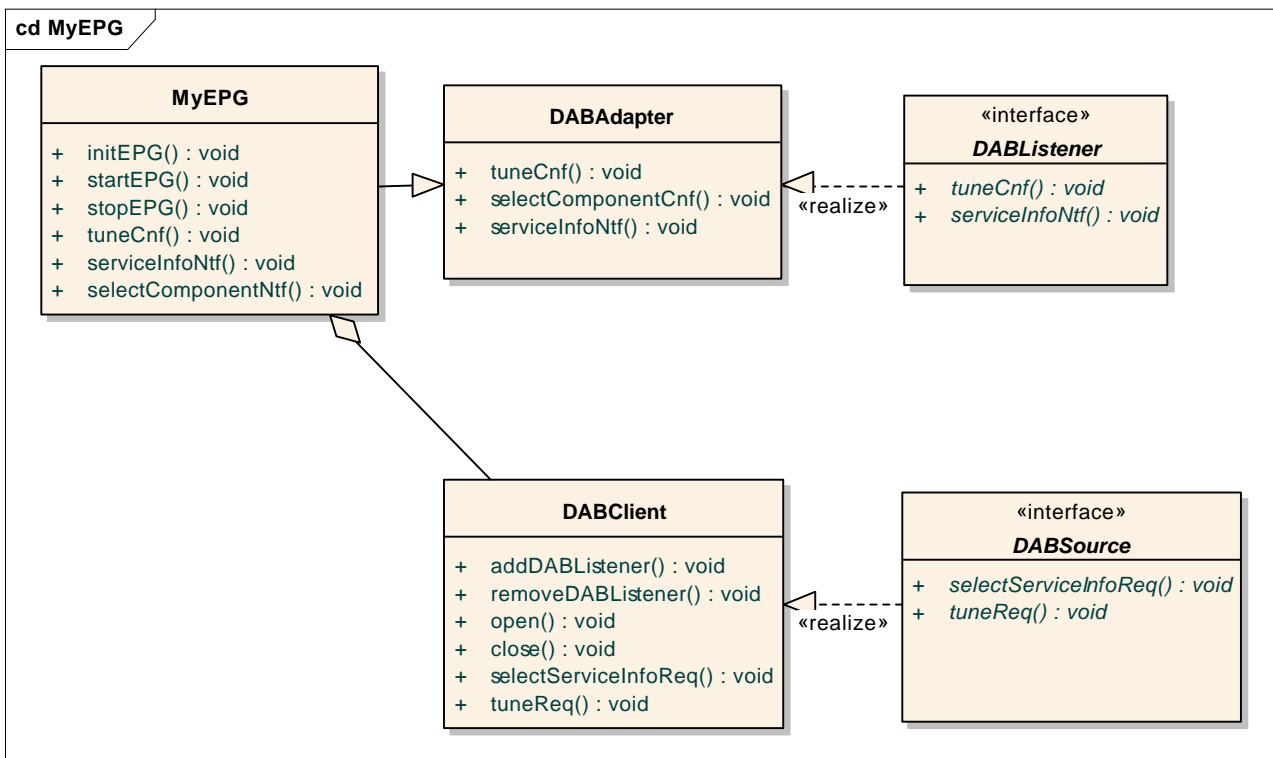


Figure 2: The classes of the EPG application

Before making any actions to the DAB receiver we have to initialize it, and register a `DABListener` for the incoming messages: in our case the EPG application class implements the listener interface by extending the `DABAdapter`.

This leads to the following initialization code:

```
public class MyEPG extends DABAdapter
{
    static public void main(String[] args)
    {
```

```

        initEPG();
        startEPG(frequency);

        pause EPG();
    }
}

```

The implementation of the main MyEPG methods are:

```

/*
 * initEPG
 * init the EPG application
 * - close connection to dab
 * - remove EPG as dab listener
 */
public void initEPG()
{
    /*
     * create dab client
     */
    DABClient dab = new DABClient();

    /*
     * register EPG as listener
     */
    dab.addDABListener(this);

    try
    {
        System.out.println("sending open request ....");

        /*
         * connect to dab
         */
        dab.open();

        System.out.println("open request sent");
    }
    catch(DABException e)
    {
        System.out.println("error on open request:"+e.toString());
    }
} // end initEPG() method

/*
 * stopEPG
 * stop the EPG application
 * - close connection to dab
 * - remove EPG as dab listener
 */
public void stopEPG()
{
    ...

    try
    {
        /*
         * close connection to dab
         */
        dab.close();

        /*

```

```

        * remove registered listener
        */
        dab.removeDABListener(this);

    }
    catch(DABException e)
    {
        System.out.println("Error during close request:"+e.toString());
    }
    finally
    {
        System.exit(0);
    }
}

```

NOTE 1: The `open()` method is synchronous, i.e. the application is blocked until the method returns.

NOTE 2: The registration procedure (`dab.addDABListener(...)`) should be done in relation to an `open()`, but absolutely before using any asynchronous methods (see the EPG example).

NOTE 3: The closing procedure is symmetric to the opening: first we close the connection to the DAB system and then we remove the registration as a `DABListener`.

After the initialization steps we implement the basic exchange of messages for controlling the audio services in the DAB ensemble.

We will focus our attention on the following set of asynchronous methods:

on the DAB side:

- `dab.tuneReq(frequency, mode);`
- `dab.selectServiceInfoReq(true, true, true, true);`

on the `DABListener` side:

- `public void selectServiceInfoCnf(SelectServiceInfoCnfEvent e)`
- `public void serviceInfoNtf(ServiceInfoNtfEvent e)`
- `public void tuneCnf(TuneCnfEvent e)`

with the events:

- `TuneCnfEvent`
- `SelectServiceInfoCnfEvent`
- `ServiceInfoNtfEvent`

After tuning to a specific frequency, we register `MyEPG` for receiving notification messages about the available services on the DAB ensemble (`EnsembleInfo`, `ServiceInfo`, `ComponentInfo`). The usage of the information received by the `DABClient` is a task that is specific to the application. Here, we demonstrate it selecting an audio component.

```

/*
 * startEPG
 * start the EPG application
 * tune dab receiver
 * @param frequency frequency to tune
 */
public void startEPG(int frequency)
{
    try
    {
        /*

```

```

        * tune to specified frequency
        */
        dab.tuneReq(frequency, DABConstants.transmissionModeAutomatic);
    }
    catch(DABException e)
    {
        /*
         * do error handling
         */
    }
    // to be continued
}

/*
 * tuneCnf
 * overload tuneCnf of DABAdapter/DABListener
 * to receive TuneCnfEvent from dab
 * @param e tune confirm event from dab
 */
public void tuneCnf(TuneCnfEvent e)
{
    /*
     * check result of tune request
     */
    int result = e.getResult();

    /*
     * get tuned frequency
     */
    int tunedFrequency = e.getTuneFrequency();

    /*
     * log progress
     */
    System.out.println("Tune cnf received,
        result = " + DABConstants.result2String(result)+"
        frequency = " + Integer.toString(tunedFrequency));
    return;
}

```

A tune request message is sent for tuning to a specific frequency (in Hz), using a specific mode (see note 4). If it is successful, a tune confirmation message is delivered and the receiver is tuned to the requested ensemble (see note 5).

NOTE 4: In the DAB specification several modes are specified for the transmission of a DAB ensemble: some DAB receivers can automatically detect the specific transmission mode of an ensemble, in others such a parameter has to be done explicitly (see DAB specification).

NOTE 5: Other more sophisticated tuning actions can be done using the scanReq() method (see DAB Java API).

```

/*
 * startEPG
 * start the EPG application
 * tune dab receiver
 * select service directory information
 * @param frequency frequency to tune
 */
public void startEPG()
{
    /*
     * continued from above

```

```

    */
    try
    {
        /*
        * select service information on ensembles, services and components
        * automatic delivery of the information objects is selected
        */
        dab.selectSIReq(true,true,true,true);
    }
    catch(DABException e)
    {
        /*
        * do error handling
        */
    }
}

/*
* selectSICnf
* overload selectSICnf of DABAdapter/DABListener
* to receive selectSICnf from dab
* @param e service info subscription confirm event from dab
*/
public void selectSICnf(SelectSICnfEvent e)
{
    /*
    * get result of selection
    */
    int result = e.getResult();

    /*
    * get selection state for ensembles
    */
    int bEnsembleInfoSelectionState = e.getEnsembleInfo();

    /*
    * get selection state for services
    */
    int bServiceInfoSelectionState = e.getServiceInfo();

    /*
    * get selection state for components
    */
    int bServiceInfoSelectionState = e.getComponentInfo();
}
return;
}

/*
* SINTf
* overload siNtf of DABAdapter/DABListener
* to receive asynchronous SINTfEvents from dab
* @param e service info notification event from dab
*/
public void siNtf(SINTfEvent e)
{
    int notificationType;
    EnsembleInfo ensembleInfo = null;
    ServiceInfo serviceInfo = null;
    ComponentInfo componentInfo = null;

    /*

```

```

    * get type of notification
    */
notificationType = e.getNotification();

switch(notificationType)
{
    case DABConstants.notificationEnsembleAdded:
    case DABConstants.notificationEnsembleRemoved:
    case DABConstants.notificationEnsembleChanged:

        ensembleInfo = e.getEnsembleInfo();
        /*
         * notify the Application of an Emsemble info
         */
        break;

    case DABConstants.notificationServiceAdded:
    case DABConstants.notificationServiceRemoved:
    case DABConstants.notificationServiceChanged:

        serviceInfo = e.getServiceInfo();
        /*
         * notify the Application of a Service info
         */
        break;

    case DABConstants.notificationComponentAdded:
    case DABConstants.notificationComponentRemoved:
    case DABConstants.notificationComponentChanged:

        componentInfo = e.getComponentInfo();
        /*
         * notify the Application of a Component info
         */
        break;

}
return;
}

```

The application uses `selectServiceInfoReq` for receiving information about all available DAB components (ensemble, services, components). After receiving a confirmation of the request, the `DABClient` will notify every change in the DAB signal information (addition, changing, and removing of ensemble, services, and components). Specifically in this case we ask to receive with the notification the information related to the particular info object (see DAB Java specification for details).

The information is delivered to the application using a special event; the usage of the carried information depends on the application strategy.

The final step for our simple EPG application is to select a particular audio component, supposed that we have collected all the information about the services and service components / components available for the selected ensemble. We assume that the user has selected a service component somehow and that the EPG has identified the selected component.

```

public void selectAudio(ComponentInfo componentInfo)
{
    if(componentInfo.getType() ==
        DABConstants.componentTypeForegroundSound)
    {
        try
        {
            dab.selectComponentReq(
                componentInfo.getId(),
                DABConstants.selectionModeReplace);
        }
    }
}

```



```

        catch(Exception _e)
        {
        }
    }

// DABListener interface
public void selectComponentCnf(SelectComponentCnfEvent e)
{
    System.out.println(
        "Result"+Integer.toString(e.getResult()));

    return;
}

```

4.3.2 Ticker

The next example, a stock market ticker, demonstrates how DAB Java applications can access data broadcast on DAB. The ticker consists of two classes - as it is displayed in figure 3. The `Ticker` class is the application's envelope, which it is responsible for the application lifecycle and the presentation of the information. The `Decoder` uses the DAB interface to receive the information and retrieve the content from the delivered data objects. `Ticker` communicates with `Decoder` uses the event-listener pattern. Additionally, it controls the lifecycle of the decoder. In the remaining part of this clause we will only show, how the decoder is implemented - as the `Ticker` class is not dependent on the DAB interface.

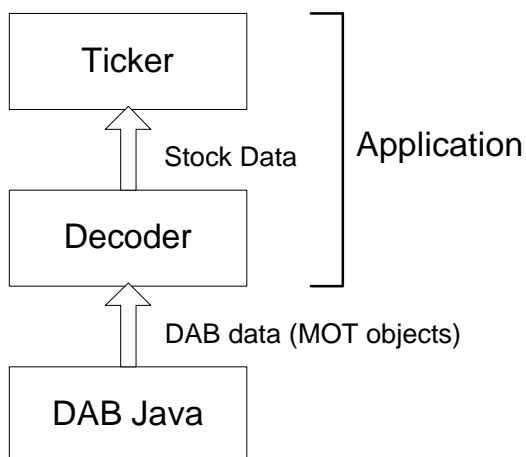


Figure 3: The architecture of the ticker

The decoder is implemented as follows:

```

import dab.*;

/*
 * Decoder class
 * handles the communication with dab
 */
public class Decoder extends DABAdapter
{
    private DABClient dabClient;
    private StockListener listener=null;
    private ComponentId componentId=null;
}

```

The variable `dabClient` contains the interface to DAB. `listener` contains the receiver of stock events (i.e. a `Ticker` object). The `componentId` specifies the identifier of the DAB-Java component which contains the stock data.

```

public Decoder()
{
    dabClient = new DABClient();
}

```

When the decoder is created, we also create a DAB client. Note, that the client is not yet configured. This means the decoder is ready for decoding, but no actual action is taken.

```

public void startDecoding(ComponentId componentId) throws Exception
{
    this.componentId = componentId;
    dabClient.open();
    dabClient.addDABListener(this);

    dabClient.selectComponentReq(componentId,
                                DABConstants.selectionModeAdd);
}

```

The decoding is started with `startDecoding`. First the DAB client is configured. The `open` call sets up the connection to the receiver. After that the decoder registers itself as a listener of DAB events (it is a subclass of `DABAdapter`, which is an adapter class of the `DABListener` interface). Then, the DAB Java component is selected using `selectComponentReq()`. We will assume that the ticker application will follow the slideshow user application model. This means objects will be delivered after the confirmation `selectComponentCnf` is sent. Here, the confirmation is ignored (the method needs not to be implemented as the default implementation in `DABAdapter` exactly behaves like that).

```

public void stopDecoding() throws Exception
{
    dabClient.selectComponentReq(serviceId,
                                DABConstants.selectionModeRemove);
    dabClient.removeDABListener(this);
    dabClient.close();
}

```

In `stopDecoding` we do the reverse setup. First the service is stopped and then the client is shut down.

```

public void addStockListener(StockListener listener)
{
    this.listener = listener;
}

public void removeStockListener(StockListener listener)
{
    this.listener = null;
}

private void notifyStockEvent(StockEvent e)
{
    if (listener != null)
        listener.stockEvent(e);
}

```

The communication with the `Ticker` object uses the event-listener model. Thus, the `Ticker` object has to register itself as a listener calling `addStockListener`. With `removeStockListener` the object will no longer receive events. Stock events are distributed using `notifyStockEvent`.

```
public void objectNtf(ObjectNtfEvent e)
{
    synchronized (this) {
        Stock[] stocks;
        try
        {
            stocks = decode(((MOTObject)e.getObject()).getBody());
        }
        catch (Exception exc)
        {
            return;
        }

        for (int i = 0; i < stocks.length; i++)
            notifyStockEvent(new StockEvent(this, stocks[i]));
    }
}
```

When `startDecoding` was called, the decoder will receive object notifications, which means that the method `objectNtf` is called from the DAB VM. First, we will decode the delivered DAB object. Note, that it needs to be cast to a `MOTObject`, because we assume the stock data is transported using the MOT protocol [1]. The result is a list of stocks that are delivered as `StockEvents` one by one.

4.4 Command Types

The commands supported by the DAB package can be categorized as follows.

- Selecting an Ensemble:
 - Tune: Tune directly to a specified frequency.
 - Search: Search for an Ensemble.
- Accessing Service Directory:
 - SelectSI: Subscribe to Service Directory information.
 - GetEnsembleInfo: Get information about a specified ensemble.
 - GetServiceInfo: Get information about a specified service.
 - GetComponentInfo: Get information about a specified component.
- Monitoring Reception Conditions:
 - SelectReceptionInfo: Subscribe to Reception Condition information.
- Selecting Components:
 - SelectComponent: Start or stop a service component carrying a user application. In case of an audio component decoding of audio samples is started automatically. In case of a data service, the service can be accessed with the `SelectObject` command.
 - SelectApplication: Launch a Java application.
 - SelectComponentStream: Get access to the data stream of the component. (e.g. packet data stream)

- Selecting Objects:
 - SelectObject: Request data objects for delivery with or without automatic updating.
- Scanning for DAB Services:
 - Scan: Scan a specified frequency range for DAB Ensembles and update the Service Directory.
- Miscellaneous:
 - OperationControl: access and modify parameters of the receiver.
 - GetLocationInfo: retrieve location information from the receiver.

In the following clauses the typical use of these command types is presented (the launch of Java applications is explained in the runtime package). Note, that in the message sequence charts the arguments of the calls do not represent actual parameters. Only qualitative information is shown to simplify the charts.

4.4.1 Tuning

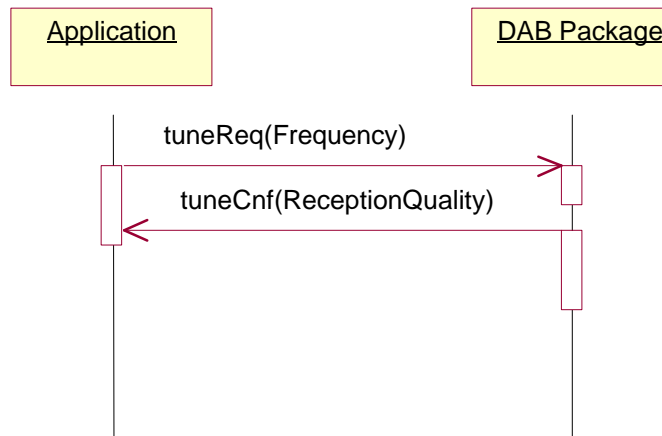
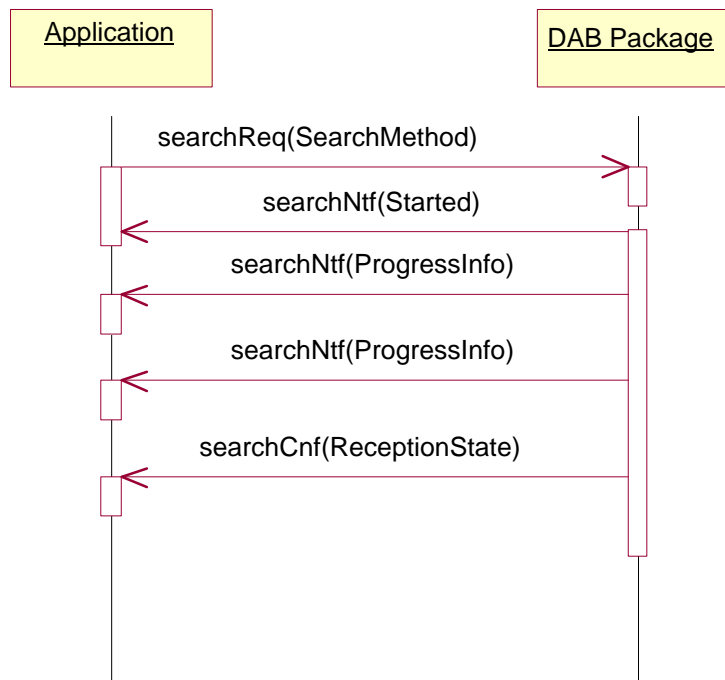


Figure 4: Tuning to an ensemble

The receiver is tuned by calling `tuneReq`. The receiver will tune to the requested frequency and respond with `tuneCnf` confirmation. The confirmation contains information about the reception quality.

4.4.2 Searching

**Figure 5: Searching for an ensemble**

To search for ensembles, the application calls `searchReq`. The package responds with a notification that the search has started. Other notifications are sent in between depending on the search method (e.g. a 16 kHz step was made). The transaction ends with a `searchCnf` confirmation containing the resulting state of the search process.

4.4.3 Scanning

Scanning means looking for ensembles on a specified range. Essentially, it is like searching except that the scanning process looks for all ensembles in the range. When the command has been issued, notification will be sent, after the scanning has been started. Further notifications are sent during the scan, which inform about the progress. When the scan is terminated, a confirmation is sent, which contains information about the scan and the state of the receiver.

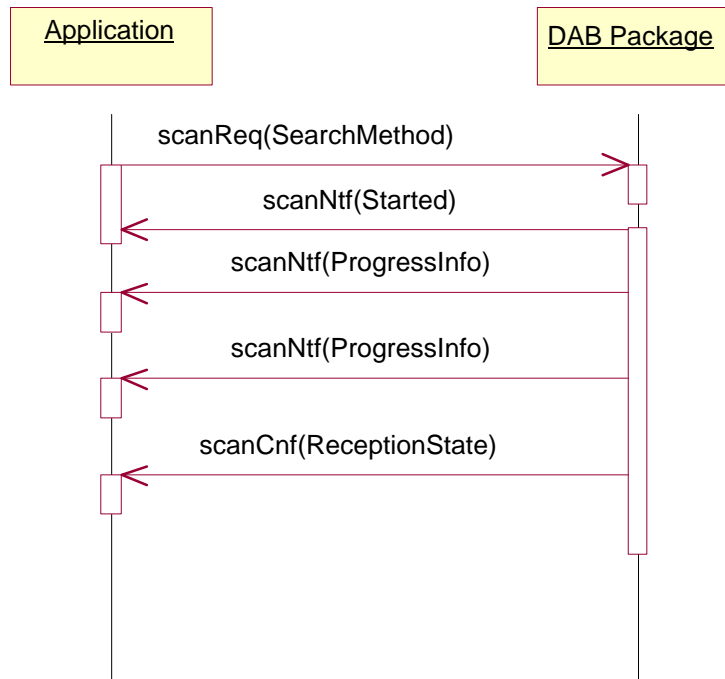


Figure 6: Scanning

4.4.4 Accessing service directory information



Figure 7: Accessing service directory information

The application, which likes to subscribe to information about the service directory of the tuned ensemble, calls `selectSIRReq` specifying the service element type.

After the confirmation is sent, the package will transmit notifications back to the application. New service elements are indicated in the notification with the flag "Added", for elements that have changed "Changed" is set and for elements that are removed "Removed" is set.

The application finishes the subscription calling `selectSIRReq`.

Note, that application can determine, whether it likes to get the respective objects of the service directory (e.g. the ensemble information) directly using this mechanism or indirectly using the other mechanism which is shown in the next clause.

4.4.5 Accessing service directory information

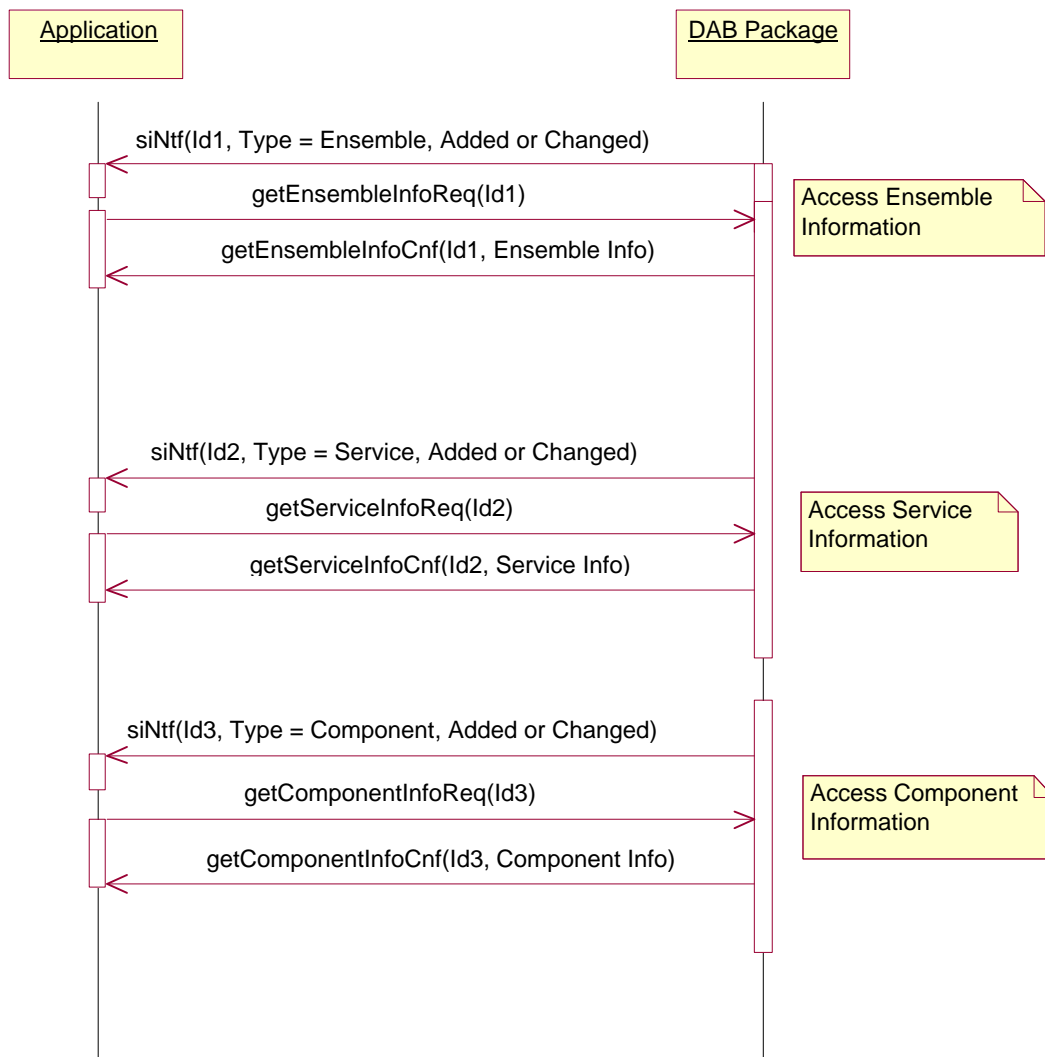


Figure 8: Accessing service directory information

Apart from getting service information directly (see the previous clause), the application can also use the `ServiceInfo` command to retrieve the respective service information objects. It has to specify the service identifier in the `siReq` request. The confirmation will then contain the requested object.

4.4.6 Monitoring reception quality

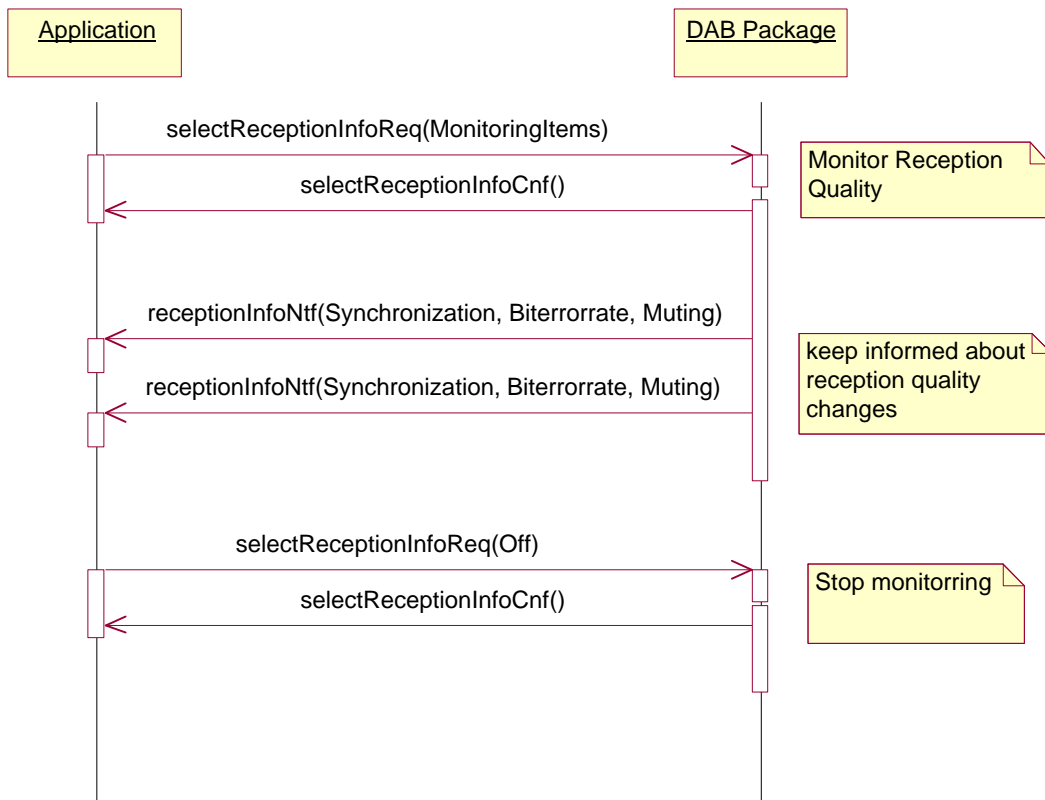


Figure 9: Monitoring reception quality

The reception quality can be monitored using the SelectReceptionInfo command. The application has to make a selectReceptionInfoReq request specifying what parameters are monitored. Then it will receive receptionInfoNtf notifications as long as the monitoring is not stopped (selectReceptionInfoReq(Off)).

4.4.7 Selecting an audio component

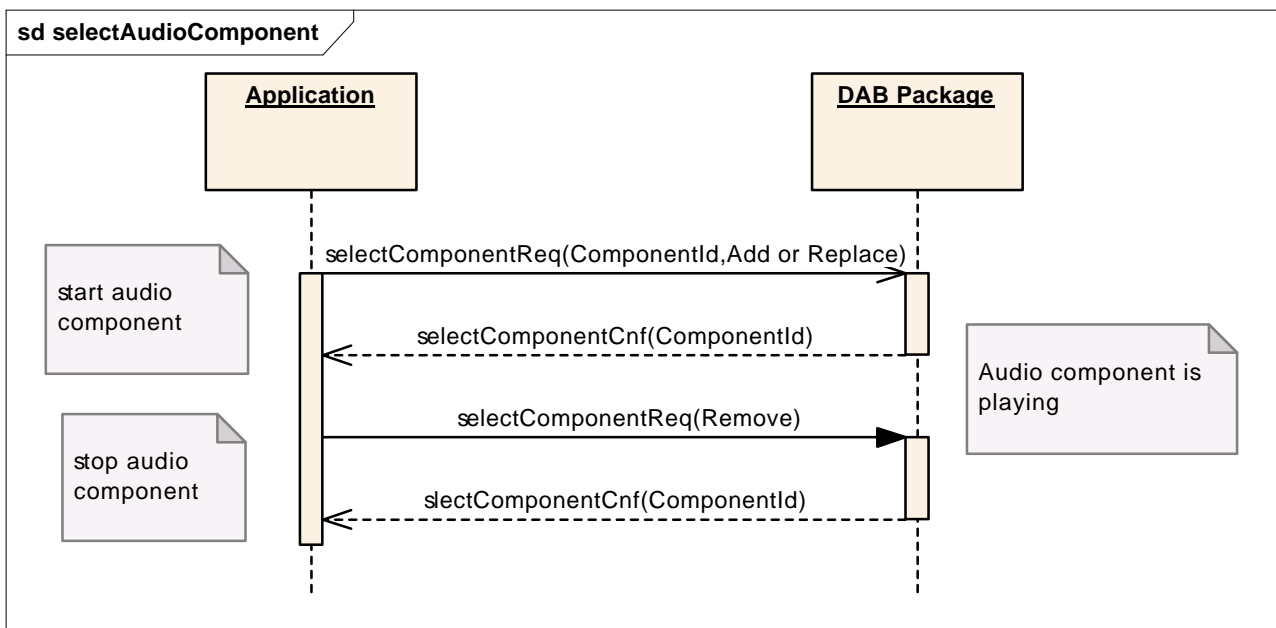


Figure 10: Selecting audio component

An audio component is started with the SelectComponent command. The application calls `selectComponentReq()` passing the identifier of the audio component. The package will start the audio service and sends back a confirmation.

To stop this audio service, the application calls `selectComponentReq` again now specifying that the component has to be removed. When the package responds with a confirmation, the audio service was stopped.

4.4.8 Selecting a slideshow or a dynamic label component

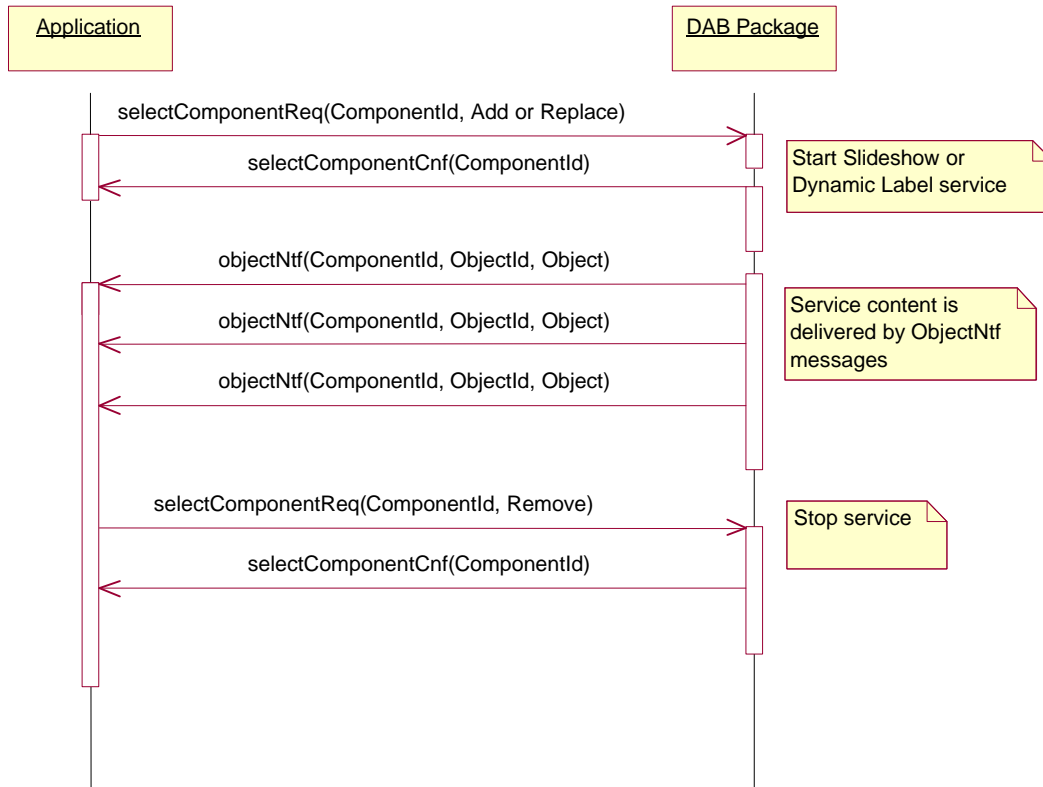


Figure 11: Selecting a slideshow or a dynamic label component

An application selects a slideshow or a dynamic label component with the SelectComponent command. When the request `selectComponentReq` with the respective component identifier is issued, the user application of the component will be started and a confirmation is sent back. The application will then receive `objectNtf` notifications containing objects of the service. To stop the service, `selectComponentReq` is called again setting `selectionMode` to `selectionModeRemove`. The removal of the service will be confirmed.

NOTE: The component identifier for the user application slideshow in a PAD service is passed as a virtual component identifier. The identifier is passed by either the `serviceInfo` object or the `siNtf` object to the controller.

4.4.9 Selecting a broadcast website component

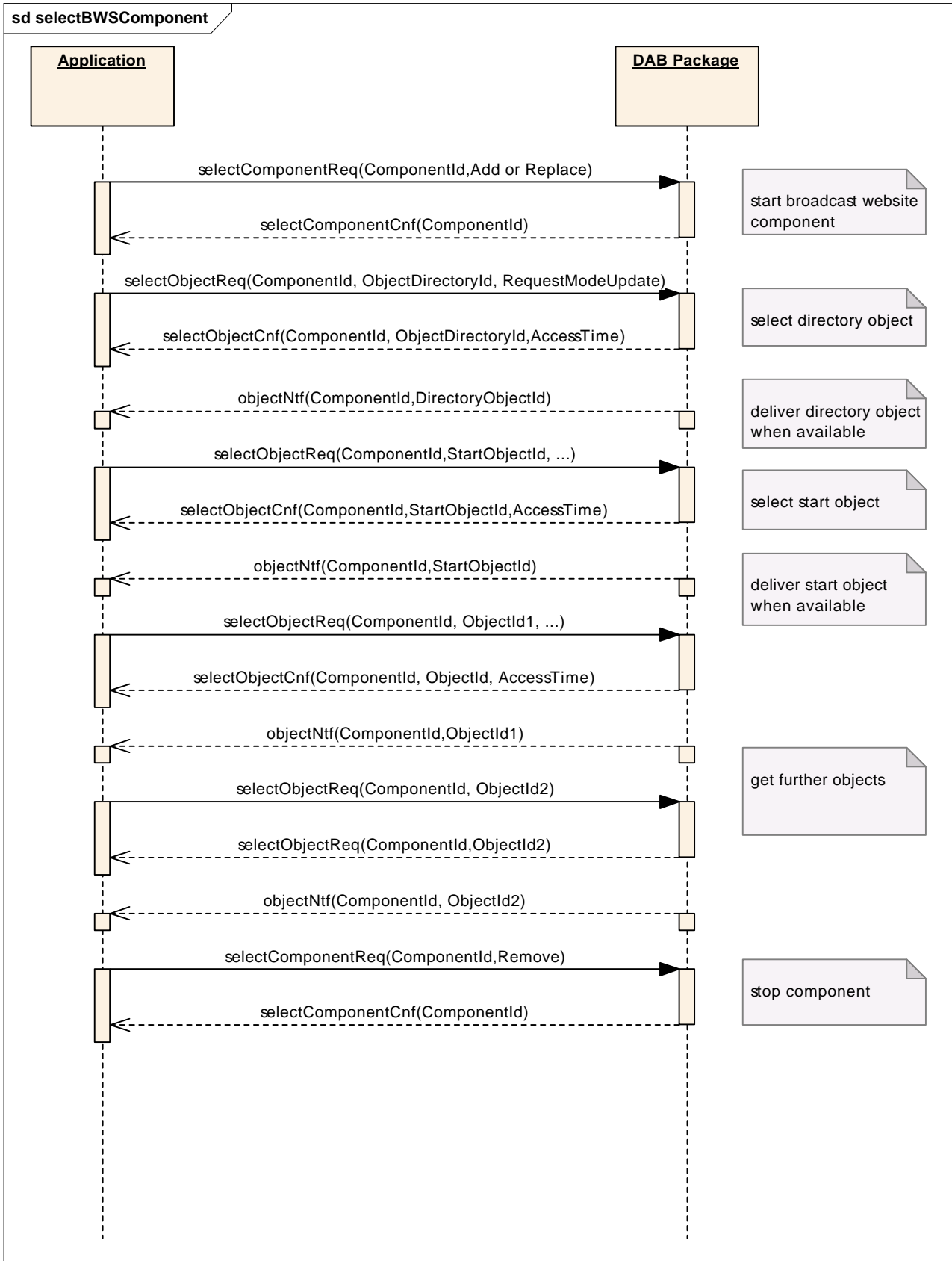


Figure 12: Selecting a broadcast website component

To run a Broadcast Website user application the component has to be selected. This is accomplished by calling `SelectComponent` command with the respective component identifier. The start of the user application will be confirmed by the package.

For a broadcast website user application at first the `DirectoryObject` is demanded by a `selectObjectReq()` with the `ComponentId` and the `objectDirectoryId` with `RequestModeUpdate`. It is recommended to use `RequestModeUpdate` to get notified by the DAB Package on the reception of a new version of the `DirectoryObject`. Next the `StartObject` is requested by a `selectObjectReq()` with the `ComponentId` and the `StartObjectId`. All further actual objects of the user application are retrieved with the `SelectObject` command. For each `SelectObject` command the DAB package will send back a confirmation including the likely access time. The actual object is received with an `objectNtf` notification.

The `ObjectDirectoryId` and the `StartObjectId` for a component could be retrieved by the `ComponentInfo` delivered by a `GetComponentInfo` command.

The component is stopped by calling `selectComponentReq()` specifying the removal of the service.

Note, that the `SelectComponent` command can be used to improve the access time of the requested time (e.g. especially caching the objects of the service).

4.4.10 Selecting an object

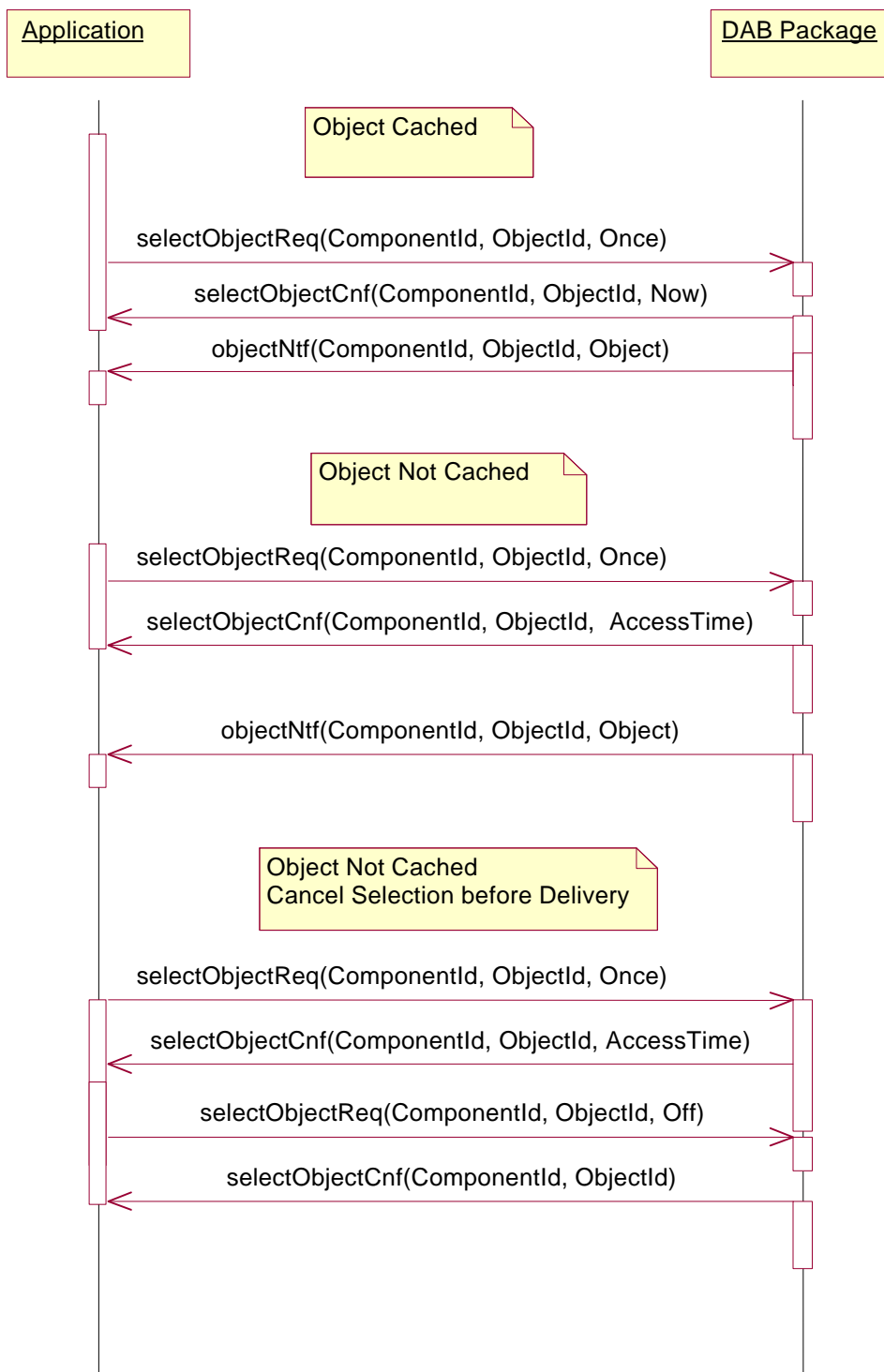


Figure 13: Selecting an object with selection mode Once

There are several cases for object selection depending on the selection mode, on the caching and on changing state between the selection modes. In figure 13 the simplest case is shown, in which an object is selected only one time.

The reaction of the package for the selection depends on whether the object is cached or not. If the object is cached, a confirmation is sent back indicating that the object is directly available. The actual object is delivered with the `objectNtf`

notification. When the object is not cached, the confirmation will indicate the access time. The application may also cancel the selection of an object in between. For that, a SelectObject command is issued specifying "Off" for the selection.

The application may wish to get updates from an object which was selected with SelectionMode=Once. This behaviour is shown in the upper half of figure 14. Because of the change of the selection mode the application has to switch off the delivery of the object.

In the lower half of figure 14, it is demonstrated, that when the application requests an object another time just once, that was not delivered yet, the object will be delivered only one time.

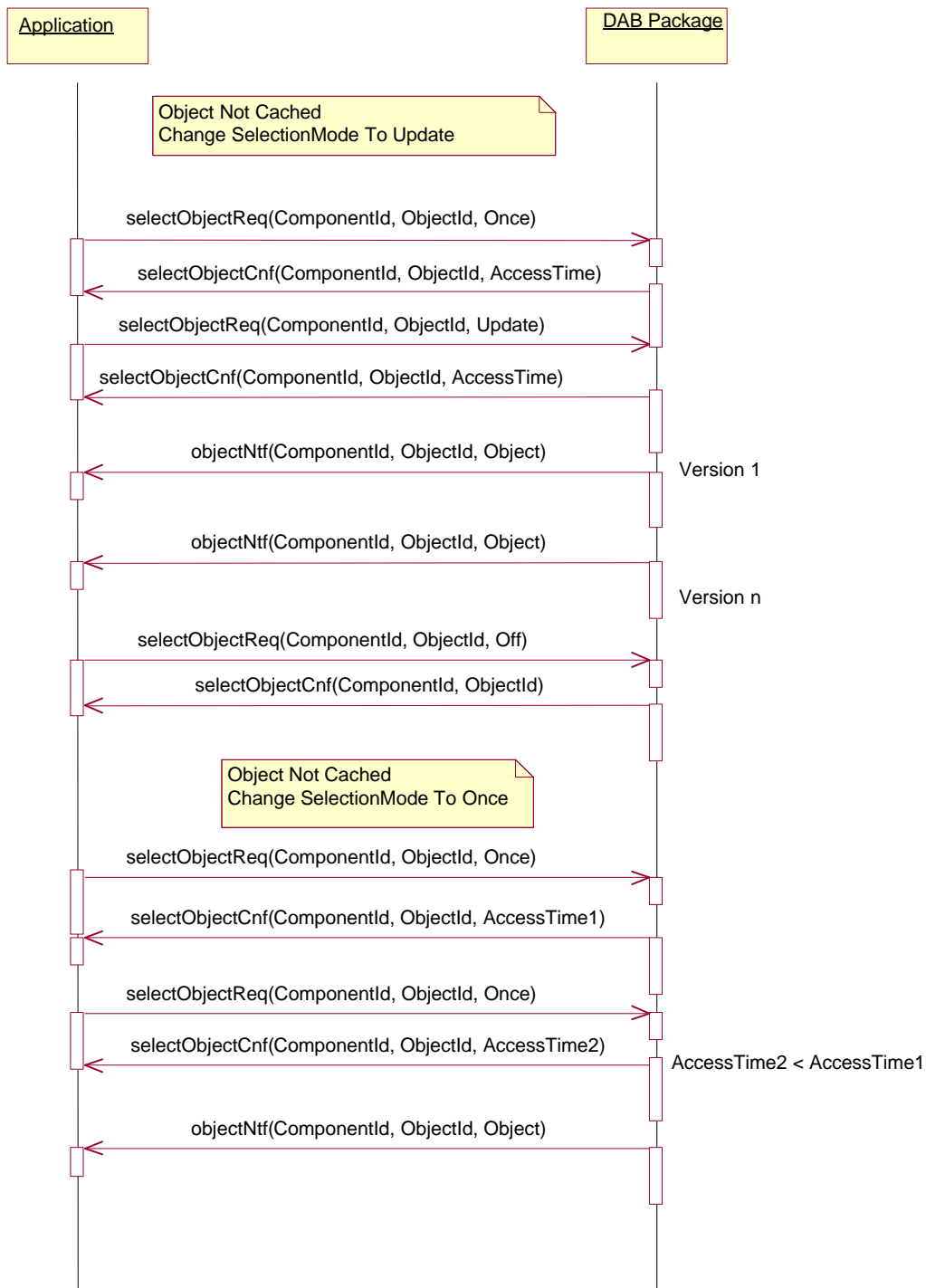


Figure 14: Selecting an object and changing selection mode from Once

Figure 15 shows the behaviour of the package, when the update selection mode was chosen. Note, that only one confirmation is sent to indicate the access time for the first version of the requested object.

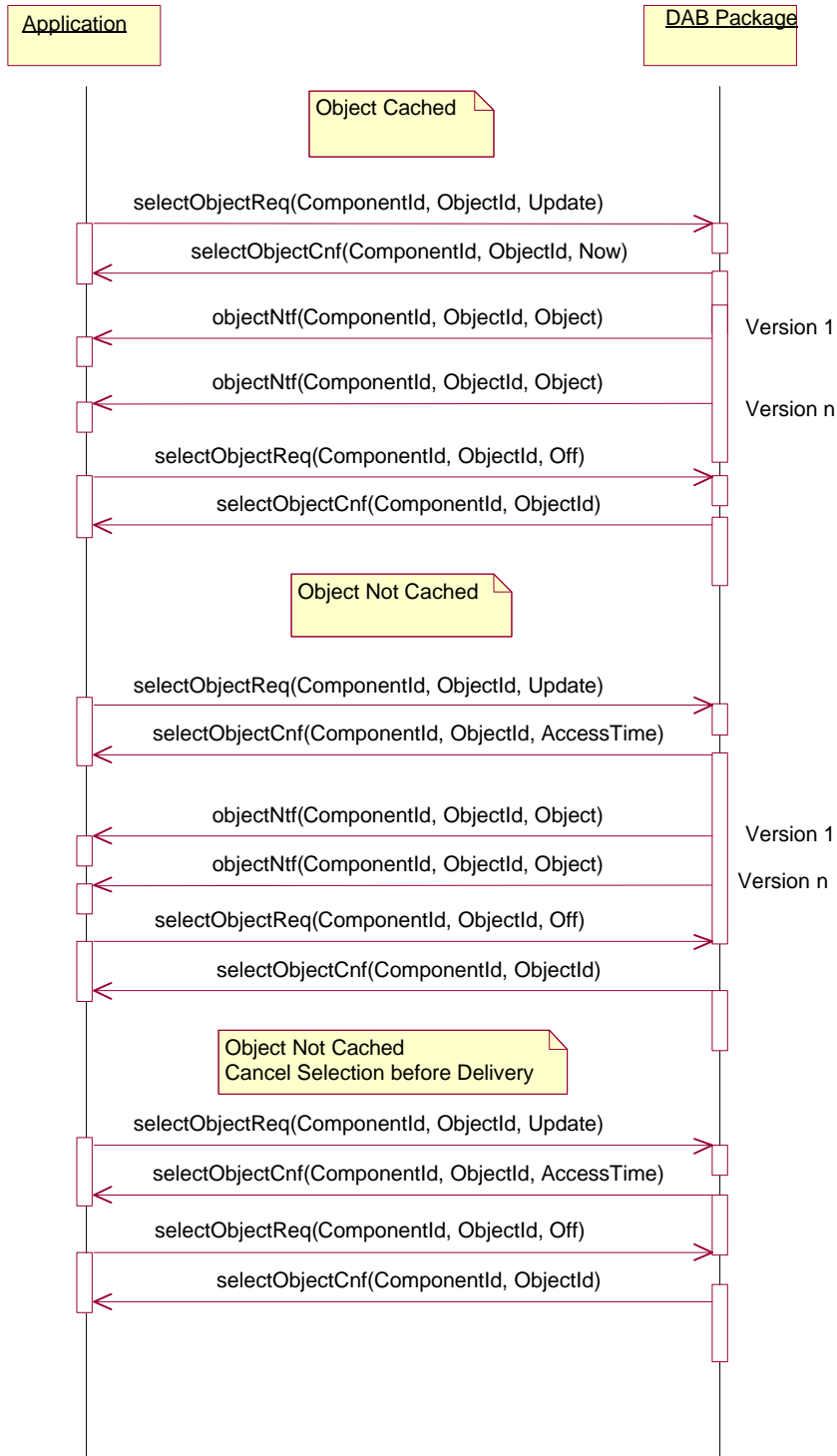


Figure 15: Selecting an object with selection mode Update

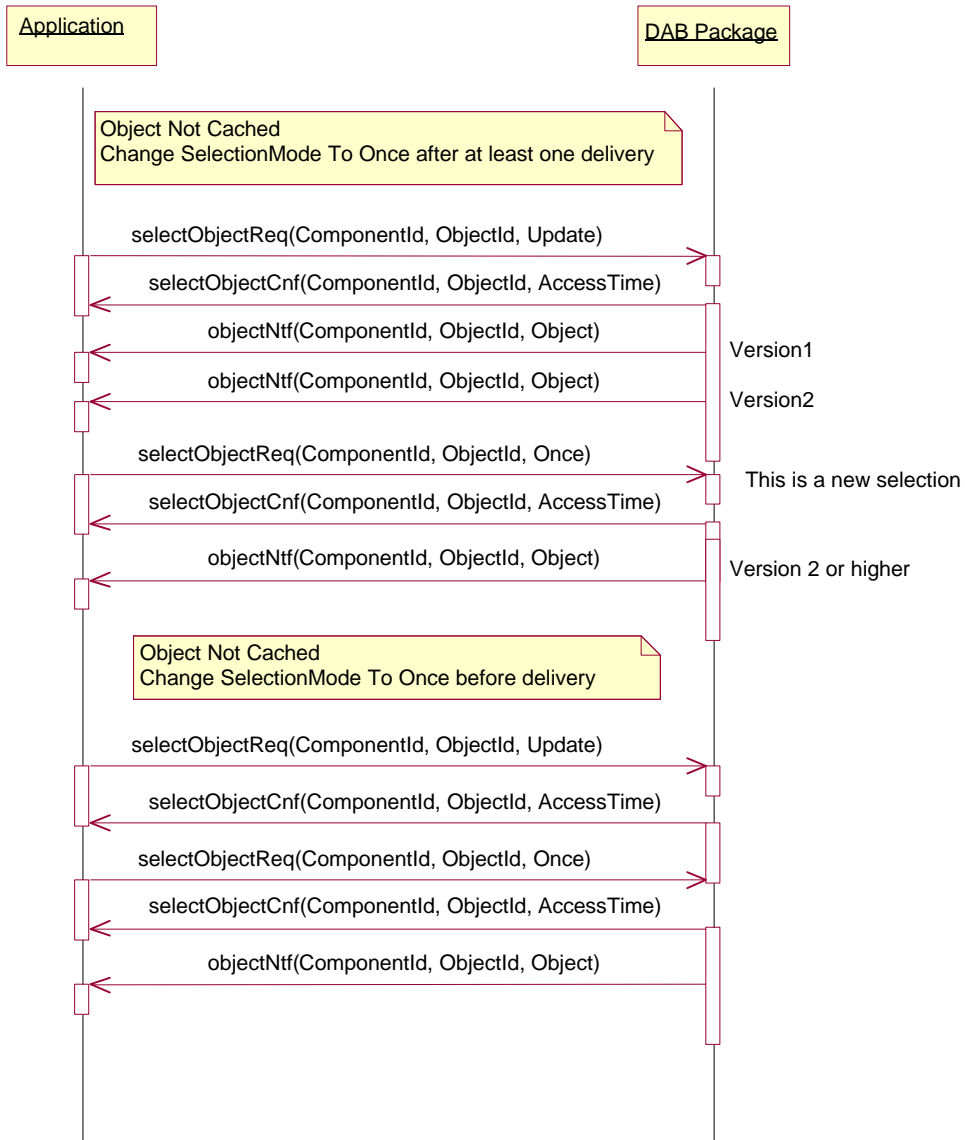


Figure 16: Selecting an object and changing selection mode from Update

In figure 16 the state change from selection mode Update to Once is shown. Note, that after the transition the object will be sent again.

4.4.11 Selecting a component stream

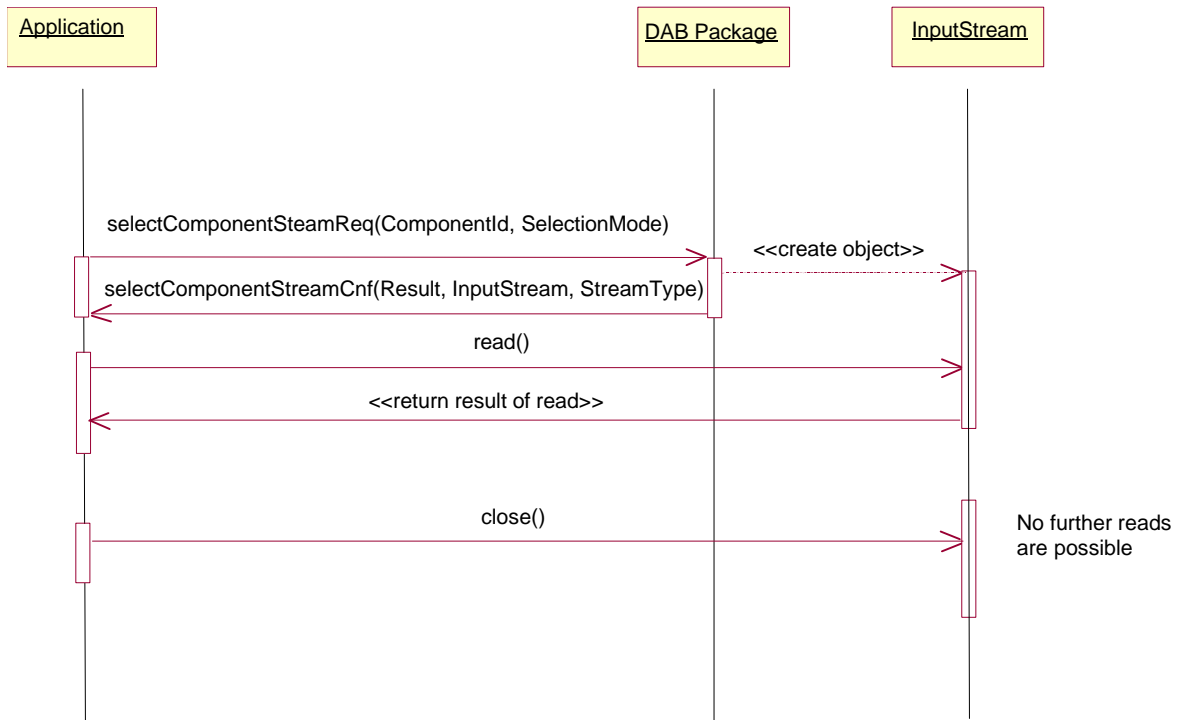


Figure 17: Selecting a component stream

Stream data can be accessed using the `selectComponentStreamReq` request. The confirmation that is sent back to the client carries an input stream object. That object provides stream data until `close` is called.

4.4.12 Operation control



Figure 18: Setting and getting operation attributes

In figure 18, it is displayed, how operation attributes are retrieved or are modified. The XXX is a placeholder for some attribute, e.g. Volume with the respective GetVolume and SetVolume operations.

In a get operation the given attribute value in the request is not considered. With regards to a set operation the given value will replace the current one, which is delivered back in the confirmation.

4.4.13 Retrieving location information

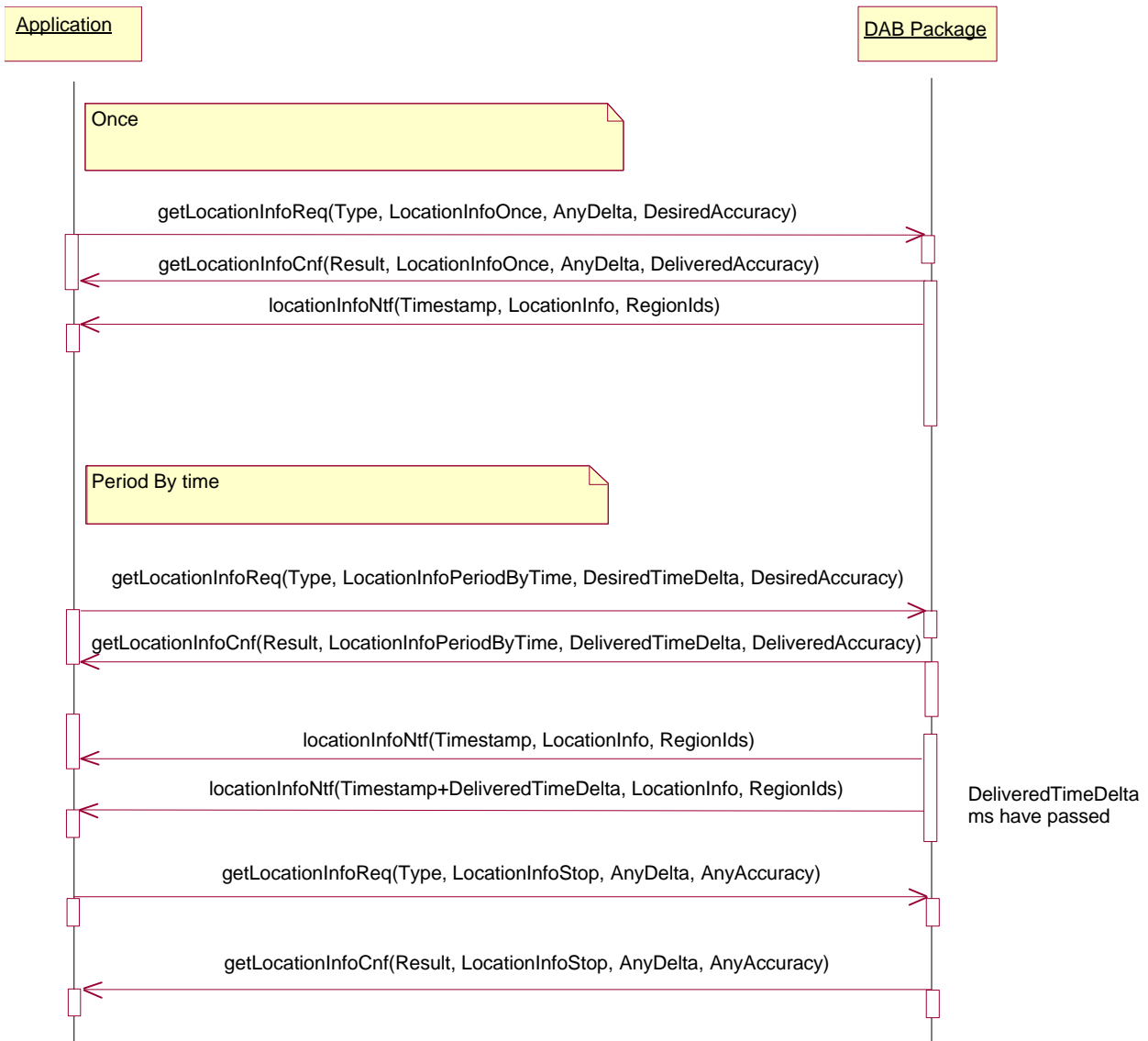


Figure 19: Retrieving location information

Location information can be retrieved in different modes. In the pull mode, LocationInfoOnce, only a single notification is sent. In the push modes, LocationInfoByTime and LocationInfoByDistance, are sent in regular intervals until the subscription is stopped.

4.5 Dependencies between the commands

Certain commands have influence on the outcome of other commands - mainly because the DAB receiver has a state. In particular:

- **Current ensemble:** If the current ensemble is modified (e.g. by the means of tune, search, or scan), all transactions which depend on this information are terminated. The exception from this rule is service information, which still can be received - even when the ensemble is changed. But as the information is not updated, only the old data is delivered (using the internal service directory).
- **Selection of components:** It is possible to select more than one data or stream component simultaneously. But only one audio component can be selected. A PAD component is only available, when the associated audio component was selected. Objects can only be requested when the respective component was selected.

4.6 Client registration

Clients have a state with respect to the used DAB receiver. They are either disconnected or connected. Transactions can only be processed in the connected state (otherwise an exception is thrown).

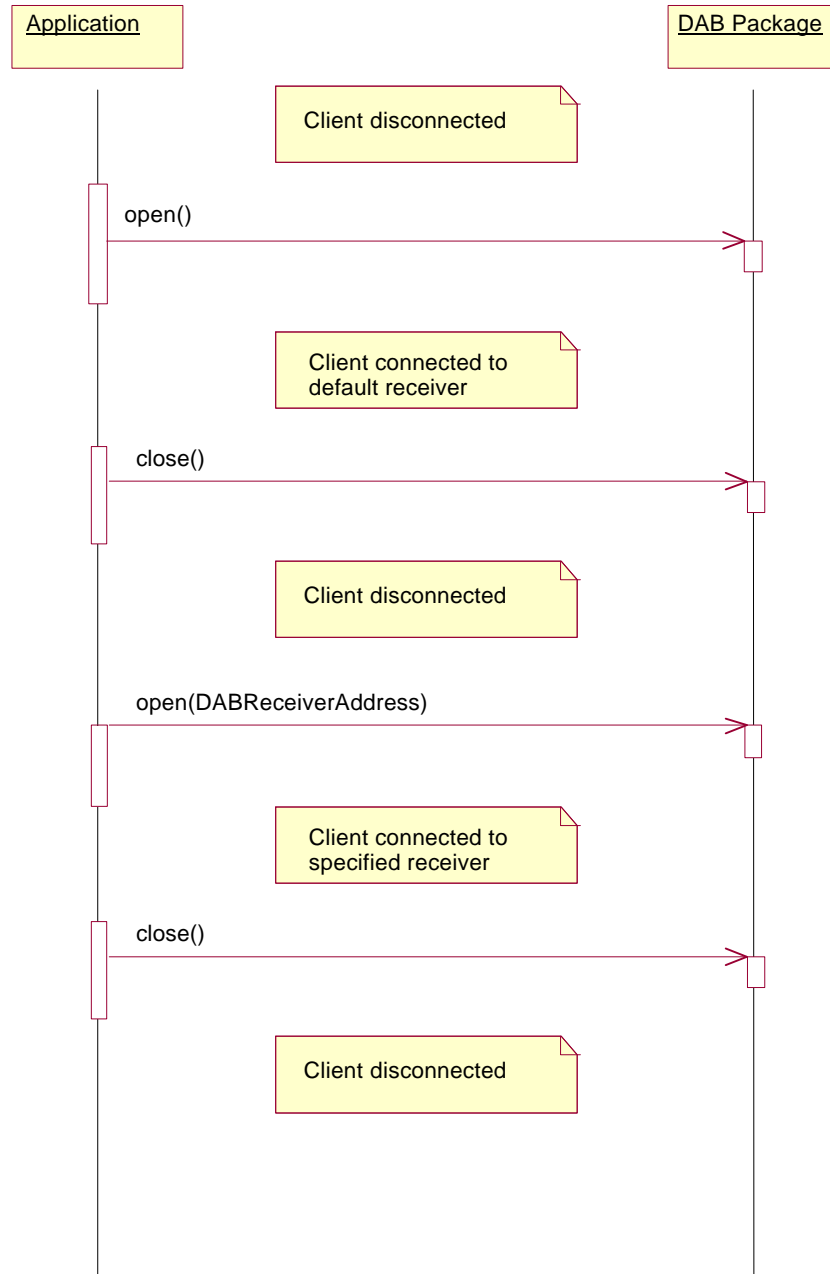


Figure 20: Client registration

A newly generated instance of `DABClient` is in the disconnected state. It is set to the connected state calling `open` in `DABClient`. It will be disconnected if `close` is called. This behaviour is depicted in figure 20. Note, that the second version only needs to be supported if multiple receivers are available.

4.7 The package structure

The classes of the `dab` package are shown in figure 21 (the details are given in the javadoc specification of the DAB Java package). Note, in contrast to the specification in the annex this and the following figures do not contain any classes or methods which are related to runtime issues (see the following clause).

`DABSource` and `DABListener` are the main classes. The `DABSource` interface contains the requests and the `DABListener` interface contains the notifications and confirmations.

Usually, these two classes will not be used directly. The `DABClient` class which implements `DABSource` is the standard interface to the DAB system. It also includes an interface to register new DAB listeners. The `DABAdapter` class provides an implementation of `DABListener` based on dummy methods, so that applets, which use only a part of the functionality of the DAB package, need not to implement all methods of `DABListener`. `DABReceiverAddress` is an abstract class that can be used to support multiple DAB tuners.

`DABException` is the superclass of all exceptions used inside the package. It is particularly used to generalize from all exceptions in `DABSource`. The `DABNotAvailableException` is thrown in the data classes when optional attributes are currently not available or not available at all. `DABConnectionException` is thrown, if there are problems with the connection to the receiver.

`DABConstants` contains all constants that are used in the rest of the package.

The package has three sub-packages: `si`, `data` and `events`. The package `si` contains classes describing service information; `data` contains various data classes that are not directly related to service information; `events` contains all event classes.

The structure of `si` is shown in figure 22b. The class `SIId` is an abstract class for referencing service information. `EnsembleId`, `ServiceId` and `ComponentId`, which are all derived from `SIId`, represent the identifier for the respective service information type. The classes `EnsembleInfo`, `ServiceInfo`, and `ComponentInfo` reflect the respective levels in the hierarchy of service information.

Figure 22a reveals the basic UML model for a DAB multiplex. In this context the class `EnsembleInfo` refer the class ensemble in the UML model. The class `ServiceInfo` present the UML class service. The class `ComponentInfo` describes one DABJava component. The DABJava component relates to the concept of service components and user applications in the following way:

- Each programme service component carries at least two `ComponentInfo` reflecting its audio service and its dynamic label service. Additionally each programm service component may carry more `ComponentInfo` describing user applications (see [3]).
- For data service components each user application is modelled as `ComponentInfo`.

In the `data` package (see figure 23) the other data classes are collected. `DABObject` is a generic class to represent all kind of data transported via DAB. `MOTObject` is a subclass of `DABObject` which models only data objects transmitted in the MOT protocol [1]. Additionally, there is derived class, `DLSObject`, for data of a DLS data service. `MOTObject` has two subclasses `MOTDirectoryObject` and `BWSObject`, which model `MOTDirectories` (in the data carousel) and objects of a BWS data service respectively. The interface `MOTObjectHeader` is used to specify information, which is provided both in `MOTObject` and in an entry in a MOT directory.

There is one more specialization, `BWSDirectoryObject`, a subclass of `MOTDirectoryObject`. This class specifies additional information in a directory, which is available only for BWS services. `BWSDirectoryIndex` is a helper class for that.

For the purpose of referencing the different kinds data objects (of type `DABObject`), `ObjectId` may be used.

The classes `Label`, `AnnouncementSupport`, `ProgrammeNumber`, `SubscriberType` and `ProgrammeType` are auxiliary classes used to model attributes inside the service information classes and `DABObject`. `LocationInfo` models location information.

Finally, we have the `events` package (see figure 24). `DABEvent` is the superclass of all events used by the package. The particular events are divided into confirmation and notification events related to the respective method in `DABListener`. Note, that there is no direct reference of `DABEvent` in `DABListener`, the associative link was only used to simplify the diagram.

Note, that there are no direct associations between EnsembleInfo, ServiceInfo, ComponentInfo and DABObject. For this the service and object identifiers are used. This means, that the service identifiers inside EnsembleInfo refer to services, the service identifiers inside ServiceInfo refer to components and so on.

To reduce the complexity of the class diagrams only the associations inside the data classes are indicated.

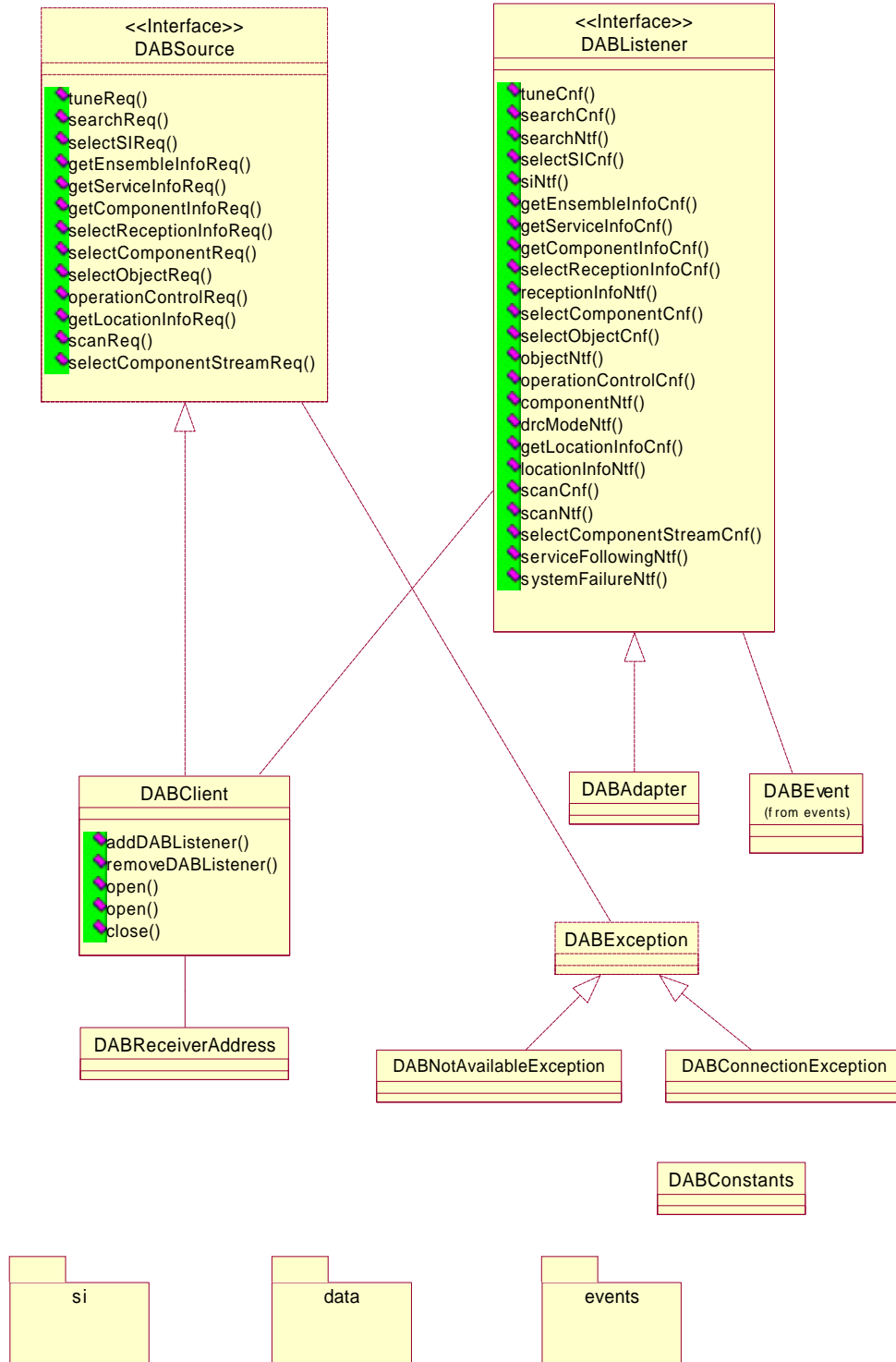


Figure 21: The classes of the dab (main) package

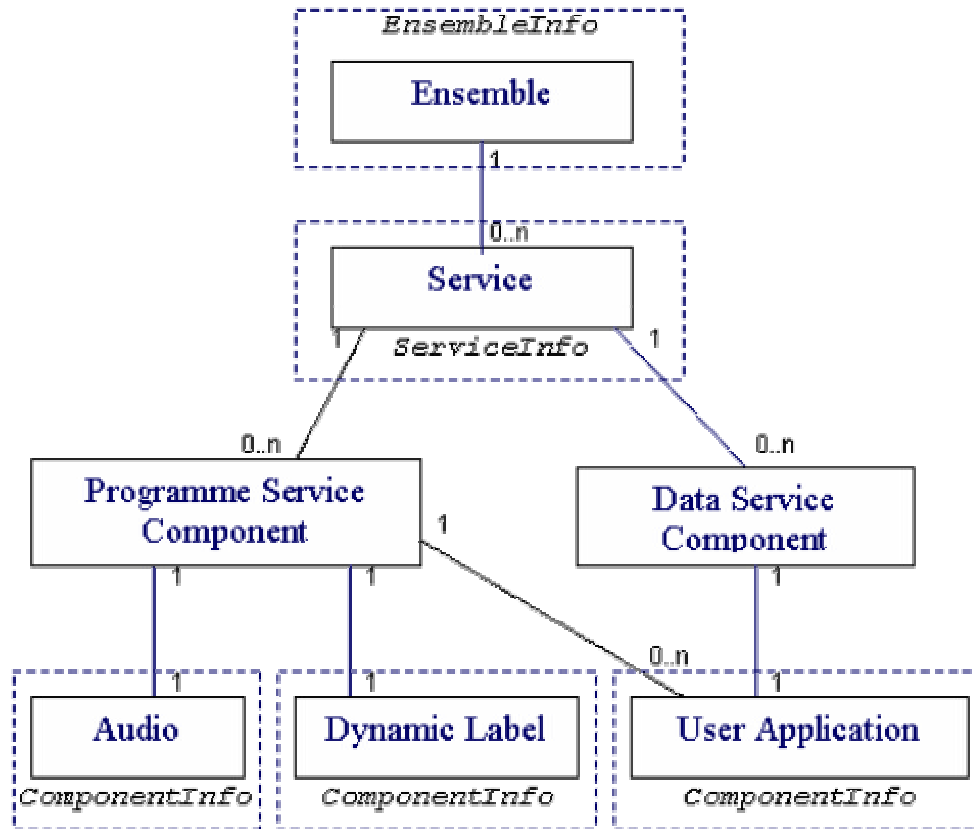


Figure 22a DAB UML Modell

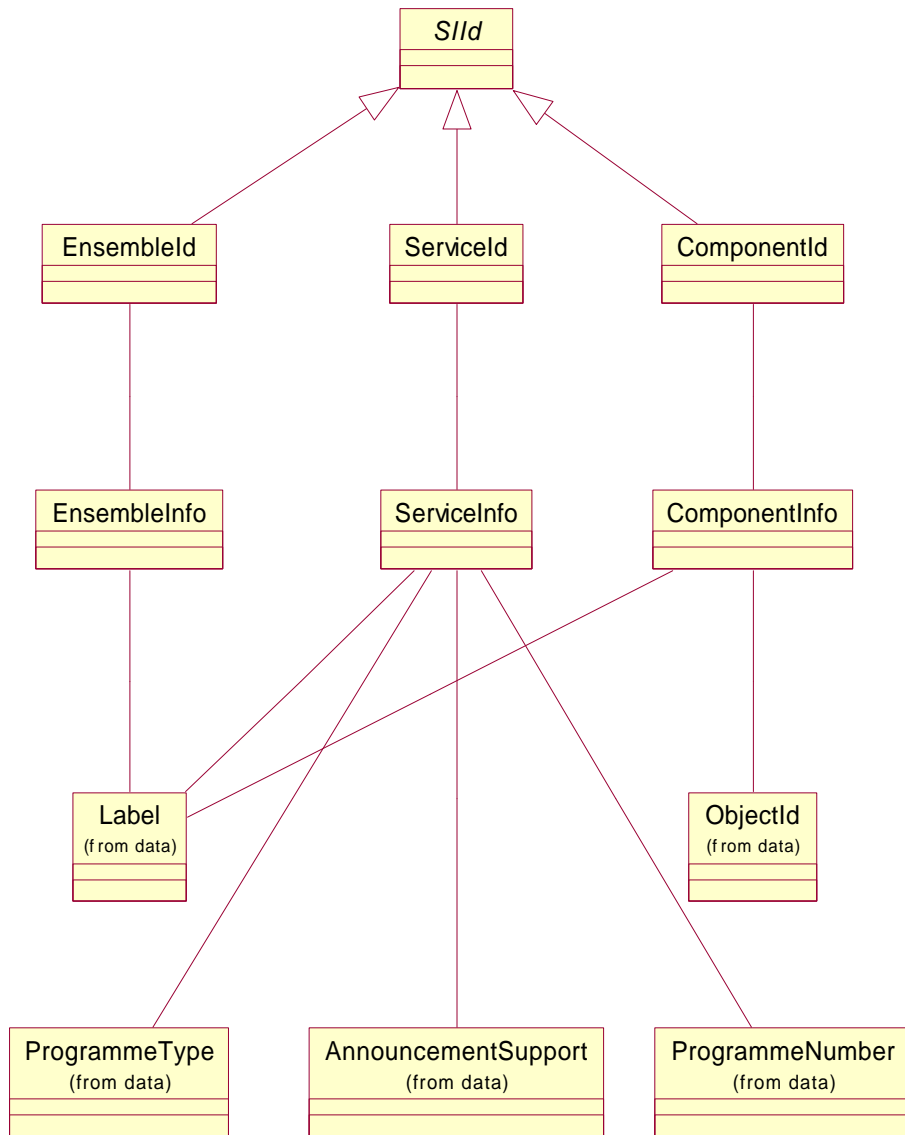


Figure 22b: The Classes of the si package

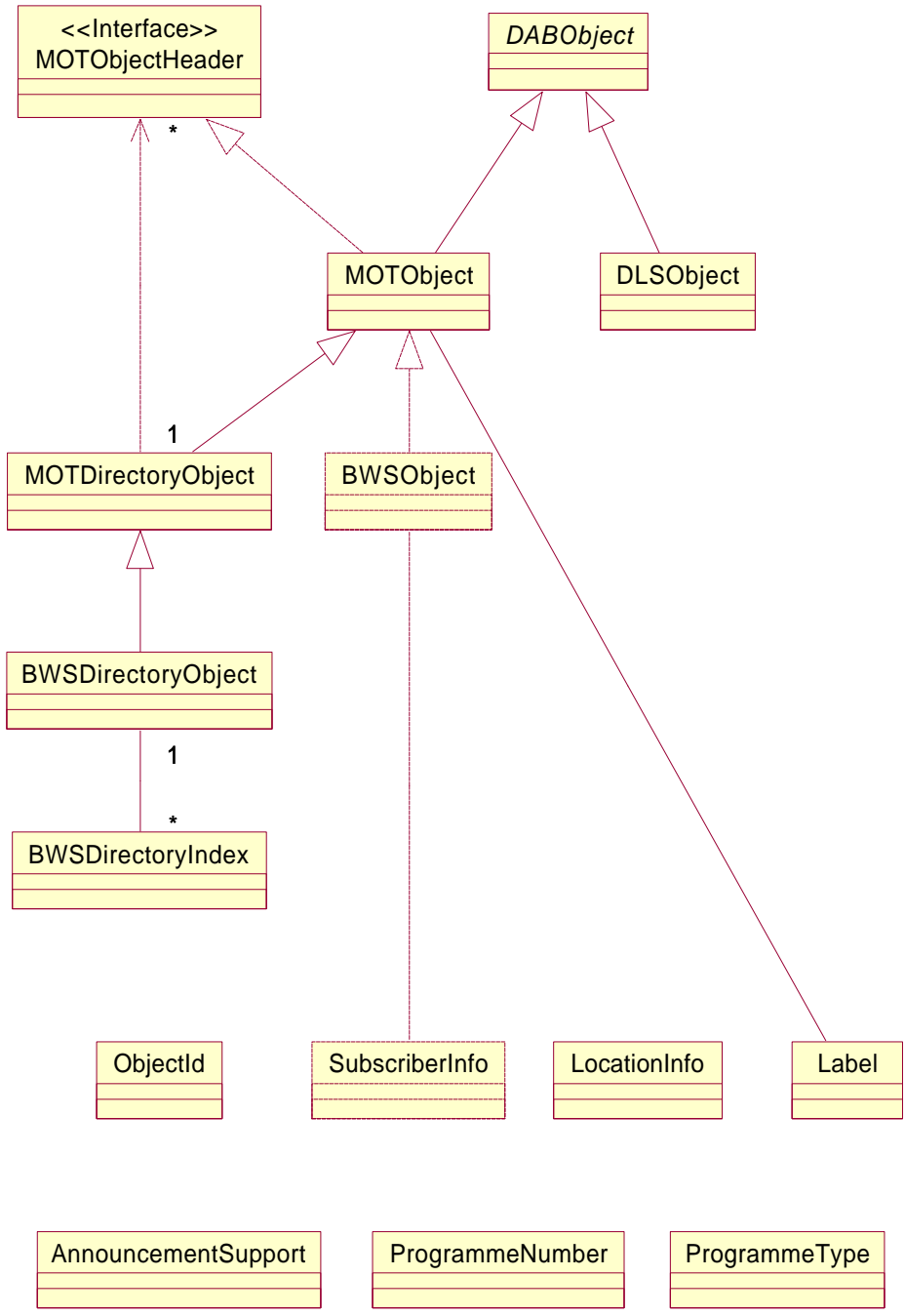


Figure 23: The classes of the data package

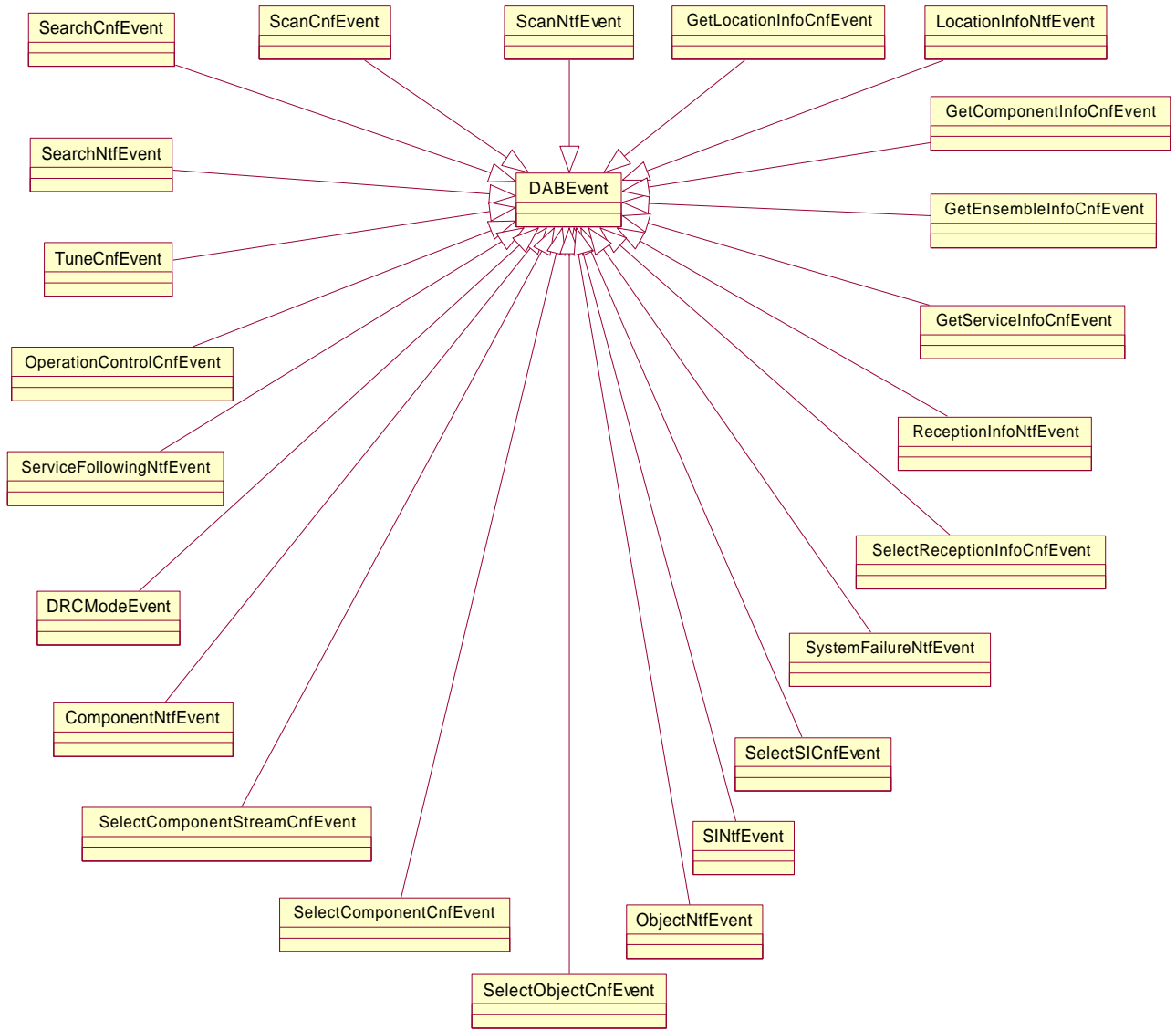


Figure 24: The classes of the events package

5 The runtime package

5.0 Summary

The DAB runtime package deals with the basic runtime components that support the execution environment for a DAB terminal. It consists of the following parts:

- **DAB application model.** This part defines a lifecycle for DAB Java applications based on the Xlet model of JavaTV.
- **Control of Java Applications.** Here, we specify how an application is transmitted, selected, launched and how its state can be controlled.
- **Security Management.** This part handles the security issues with regard to DAB Java applications.
- **Resource Management.** The resource management provides mechanisms for sharing resources between different DAB Java applications.
- **Configuration Management.** This part deals with handling of the internal profile (i.e. the profile information that is available to the application).

5.1 The DAB Application Model

The Xlet model from Java TV (see Bibliography, "DVB Java Specification") is used for all applications, which are downloaded via DAB. The state machine of the Xlet is described in the following state model (see figure 25). The application controller can control the Xlet component calling specific methods from one side and the Xlet can notify the application controller about the changes of his internal state (see the class diagram in figure 26). After the Xlet code is loaded, it can be instantiated (calling the new operator). When `initXlet()` is called, the method should initialize all the resources needed by the Xlet and put the Xlet in the PAUSED state. The life of the Xlet then is controlled calling respectively the `startXlet()` and `pauseXlet()`. Additionally, the Xlet can notify the host application about the internal state changes using the XletContext (see figure 26): for example, if no further computation is possible, the Xlet can go in the PAUSED state and can notify with the paused methods the host application.

The final step of the lifecycle of the Xlet is reached, when `destroyXlet()` method is called. The Xlet then has to deallocate all used resources.

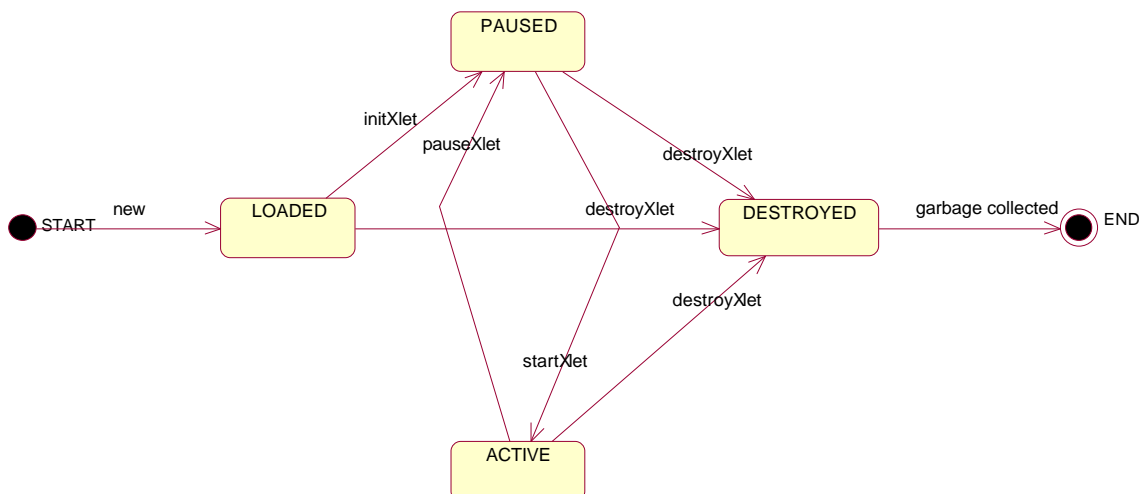


Figure 25: The Xlet Model

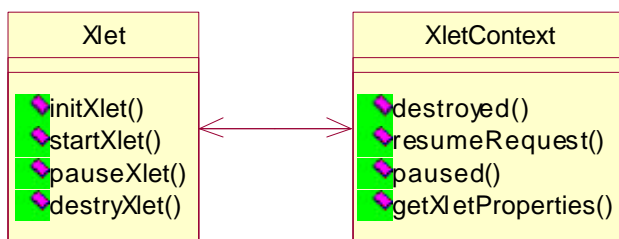


Figure 26: Context for the Xlet model

Moreover application designers are supposed to check the host environment to control if the needed resources are accessible: graphical interfaces, display area, DAB specific commands, etc. In the Xlet implementation this action is performed passing an array of strings to the loaded component using the XletContext `getXletProperties(String)`, where all the necessary variables are parameterized (see clause 5.5).

The following example demonstrates the some of Xlet methods. It is the main class of the stock market ticker (see above):

```

/*
 * class Ticker a DAB-Java application (Xlet)
 * select Java application from dab java runtime environment
 */
public class Ticker implements dab.xlet.Xlet
{
    int                m_status = INIT;
    dab.si.ComponentId m_compId = null;
    java.awt.Panel     m_panel = null;

    /*
     * methods to be implemented from the dab.xlet.Xlet interfac
     */

    /*
     * initXlet
     * is started after xlet loaded from DAB
     * @param ctx xlet container passed for dab package
     */
    public void initXlet(XletContainer cxt)
    {
        /*
         * read component id from xlet context
         */
        m_compId = (ComponentId)
                    cxt.getXletProperty("dab.xlet.componentId");

        /*
         * instantiate decoder
         */
        decoder=new Decoder(m_compId);

        /*
         * read container panel from xlet context
         */
        m_panel = (java.awt.Panel)
                    cxt.getXletProperty("dab.xlet.panelContainer");

        containerPanel.add(this);

        m_status=PAUSED_AFTER_INIT
    }

    /*
     * startXlet
  
```

```

    * is started after xlet is initialized
    */
public void startXlet() throws XletStateChangeException
{
    if (m_status==PAUSED_AFTER_INIT)
    {
        requestForStop = false;
        thread = new Thread(this);
        thread.start();
        decoder.addStockListener(this);
        decoder.startDecoding();
        started = true;
    }
    else
    {
        thread.resume();
    }

    m_status=ACTIVE
}

/*
 * pauseXlet
 * is started after xlet loaded from DAB
 */
public void pauseXlet() throws XletStateChangeException
{
    if (m_status==PAUSED_AFTER_INIT)
    {
        requestForStop = false;
        thread = new Thread(this);
        thread.start();
        decoder.addStockListener(this);
        decoder.startDecoding();
        started = true;
    }
    else
    {
        thread.resume();
    }

    m_status=PAUSED
}

/*
 * destroyXlet
 * is started after xlet loaded from DAB
 */
public void destroyXlet() throws XletStateChangeException
{
    if (m_status==PAUSED_AFTER_INIT)
    {
        requestForStop = false;
        thread = new Thread(this);
        thread.start();
        decoder.addStockListener(this);
        decoder.startDecoding();
        started = true;
    }
    else
    {
        thread.resume();
    }
}

```

```

    }
    m_status=DESTROYED
}

```

5.2 Control of Java applications

A java based user application is started on request of the user by the application launcher of the receiver. Therefore an instance of the signalled Main-Class of the application is created. The started application is allowed to start further applications available from this service. The Java application can be distributed as a set of packages, which are transported using the MOT protocol [1]. Such an application is described by a profile, which specifies the required platform, the application model and the application type. If an application is chosen for execution, the launch procedure is initiated: the procedure consists in loading the classes (archive), in setting up a runtime context to the application and in initializing the application itself.

In the following clauses these different aspects are described.

5.2.1 Packaging

A Java-based application is distributed as a sequence of Java archives, which contain a collection of classes in the JAR format (see Bibliography, "JAR Archive Documentation"). Apart from object code such a jar file can also contain arbitrary data files. Additionally, a so-called Manifest file can be used for configuration issues (see clause 5.5).

The Manifest file inside a Jar archive contains meta-information about the content of the archive (see Bibliography, "Manifest"). With regard to the launching procedure the following attributes are used:

- **Class-Path**: This attribute specifies a sequence of relative URLs, which refer to other distribution objects. These paths will be used to resolve dependencies when loading classes from the archive. Each URL has to correspond to an identifier of a MOT object.
- **Main-Class**: If this attribute is set, the application manager will ask the user for loading and executing the designated class (the name has to be a relative URL).

The following information can also be included inside the Manifest file for describing the content and for repeating attributes available at the FIC level (these values should not be used for system settings):

- **Platform**: the supported platform (see clause 6.3)
- **Access**: the access pattern (see clause 5.3)

The respective type for the MOT object is 6="system" and the subtype is 1="Java" (see [1]).

5.2.2 Transmission as DAB-Java MOT carousel

A single Java based user application is transmitted as a service component of Type "Java" (see clause 6).

The sequence of Java archives of the packaged user application is transmitted as a MOT directory (see [1]). The respective Type for each MOT object is "system" and the subtype is "java".

The following clauses describe the used MOT parameter for transmission

5.2.2.1 MOT parameter for individual objects

The following MOT parameter for individual objects are applied in DAB-Java MOT carousels

Parameter	Specified in	Mandatory for UA provider	Occurrences
ContentName	MOT	Yes	single
UniqueBodyVersion	MOT	Yes	single
Consistency	MOT	No	single
CompressionType	MOT	No	single
ProfileSubset	MOT	No	single
CAInfo	MOT	No	single
SubscriberInfo	MOT	No	multiple

Table 2: Use of MOT parameters for individual objects

For a detailed description of the usage of the parameters see MOT specification [1].

5.2.2.2 MOT parameter for the entire carousel

The following MOT parameter for the entire carousel are applied in DAB-Java MOT carousels

Parameter	Specified in	Mandatory of UA provider	Occurrences
DirectoryCompression	MOT	No	Single
DefaultConsistency	MOT	No	Single
ApplicationArchive	VM	Yes	multiple
ClassPath	VM	Yes	multiple

Table 3: Use of MOT parameters for the entire carousel

For a detailed description of the usage of the parameter see MOT specification [1].

5.2.2.2.1 The ApplicationArchive Parameter

The `ApplicationArchive` Parameter is used to indicate the relative URL of the main application archive within the directory of the service component. The `ApplicationArchive` could be specified for each version of a platform.

Syntax	Size	Type
<code>ApplicationArchive_parameter_data_field(){</code>		
Platform	8 bits	uimsbf
Version	8 bits	uimsbf
for(i=0;i<N;i++){		
ApplicationURLByte	8 bits	uimsbf
}		
}		

Table 4: The syntax of the `ApplicationArchive` parameter data field is given in Table

platform: This field identifies the platform profile for which the identified object is the application archive

version: This field identifies the version of the platform

applicationURLbyte: These fields define the relative URL of the application archive in the directory.

The `ApplicationArchive` parameter is identified by the `ParamId` value 0x25 (tbc.).

5.2.2.2.2 The Class-Path Parameter

The `Class-Path` Parameter is used to indicate the Class Path relative to the directory of the service component. The `Class-Path` could be specified for each version of a platform.

Syntax	Size	Type
ClassPath_parameter_data_field(){		
Platform	8 bits	uimsbf
Version	8 bits	uimsbf
for(i=0;i<N;i++){		
ClassPathByte	8 bits	uimsbf
}		
}		

Table 5: The syntax of the Class-Path parameter data field is given in Table

platform: This field identifies the platform profile for which the identified object is the application archive

version: This field identifies the version of the platform

classPathByte: These fields define the class path relative to the directory of the service component.

The Class-Path parameter is identified by the ParamId value 0x26 (tbc.).

5.2.3 Selecting an Application

The application manager of the DAB-Java terminal offers all java based user applications of the service for selection by the user.

The selected application is allowed to start further java based user applications from the selected service.

5.2.4 Starting an Application

If the Main-Class of an archive could be determined the application manager will execute the designated class on selection by the user. Before launching the application the complete signalled Class-Path of the application has to be available.

5.2.5 Terminating an Application

Each java application can be terminated on users request by the application manager of the DAB-Java terminal. To terminate the application `destroyXlet()` is called by the application manager on the `dab.Xlet` interface of the instance object.

A application should be able to notify that it is stopped by finishing its execution and informing the application manager through the `destroyed()` method on the `dab.XletContext` interface

5.2.6 Loading classes

The class loader used in the runtime package has to support the loading of classes transported via DAB (see clause 5.2.1). When a non-system class is referenced, its name (including the package path) is mapped to a relative URL (e.g. MPEG.Decoder is mapped to MPEG/Decoder.class). Then, the respective class file is searched first in the current distribution object. If it is not found, the search is continued in the list of distribution objects designated by the Class-Path attribute in the Manifest file (continuing from left to right). The search will stop, when the class file is found or all distribution objects were tried.

Additionally, the class loader has to take care for updates. In general, the class loader will only consider the newest version of a distribution object. If an update of such a distribution object happens during the loading of classes, the class loader shall stop the loading of the current class and restart the process for all classes of the application using the newest version.

5.2.7 Control of applications

If an application is chosen for execution, the controller will use its `DABClient` object to load the application. The result of the operation is a proxy object, which has methods for controlling the application state.

5.2.7.1 Application context

The class `DABClient` provides a method for loading applications (`selectApplicationReq`). This method uses the given user application context and id for the start object to load the application. When the transaction is completed, a `SelectApplicationCnfEvent` is sent to the listener. When the loading was successful, a proxy object is returned in the event.

5.2.7.2 Proxy

Applications are controlled (indirectly) using a simplified version of the `org.dvb.application.AppProxy` interface (from the MHP specification [2]). The concept is an indirect control of applications. This means if for example an application likes to stop another application, it calls the respective method in the `AppProxy` interface. The application controller, which implements the interface, will then call the method in the application.

The `DABAppProxy` interface offers methods to change the state of the controlled application. Each time it is checked, whether the application is permitted to issue the request for a state change. Additionally, there are methods to be notified about state changes.

For the notification of state changes, the classes `dab.events.AppStateChangeEvent` and `dab.AppStateChangeListener` are used.

The `AppStateChangeListener` class was modified to comply with the convention to indicate the event source. For that an additional interface `AppStateChangeEventSource` was defined. Finally, there is no application identifier (the reference to the proxy object can be used for that purpose).

There are six states defined for the application proxy (as it is shown in figure 28). The proxy is in the `NotLoaded` state when it is created and returned by `selectApplicationCnf` to the application. This means that the requested application was not yet (down-)loaded. When `load` is called the application will be loaded and the state of the proxy is now `Loaded`. The proxy move on to `Inited` when `init` is called. As part of the state change `initXlet` is invoked in the loaded application.

The `pause` action is possible for two states. If the proxy is in the `Inited` state, the proxy is just moved to the `Paused` state. If it was in the `Started` state, additionally `pauseXlet` is called.

The `start` action is valid for the first four states in figure 4. It is effectively a composite action, whose effect depends on the state of the proxy:

- 1) `NotLoaded`: `load`, `init`, `pause`, `resume`
- 2) `Loaded`: `init`, `pause`, `resume`
- 3) `Inited`: `pause`, `resume`
- 4) `Paused`: `resume`

The `resume` action can only be invoked, when the proxy is in `Paused`. It invokes `startXlet` and sets the proxy to `Started`. The proxy can be set to the `Destroyed` state from any other state using `stop`. During the state change `stopXlet` is called. Each state change is signalled to all `AppStateChangeEvent` listeners accordingly.

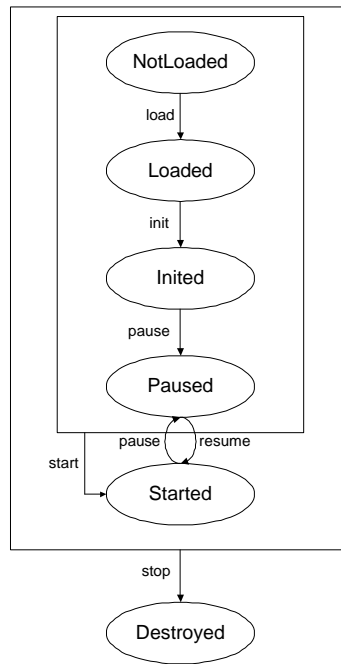


Figure 28: Proxy states

5.2.7.3 Example

In figure 28, it is shown how some application is loaded and how it gets controlled. First, the controller (here an EPG) has to select the component. In the second step the application is loaded. In the third step the application is started (an example for control). In the fourth step the EPG will register for state changes. In the fifth step it receives a state change event. Note, that the messages that are sent to and from the application container are labelled with stereotypes, as they depend on the implementation.

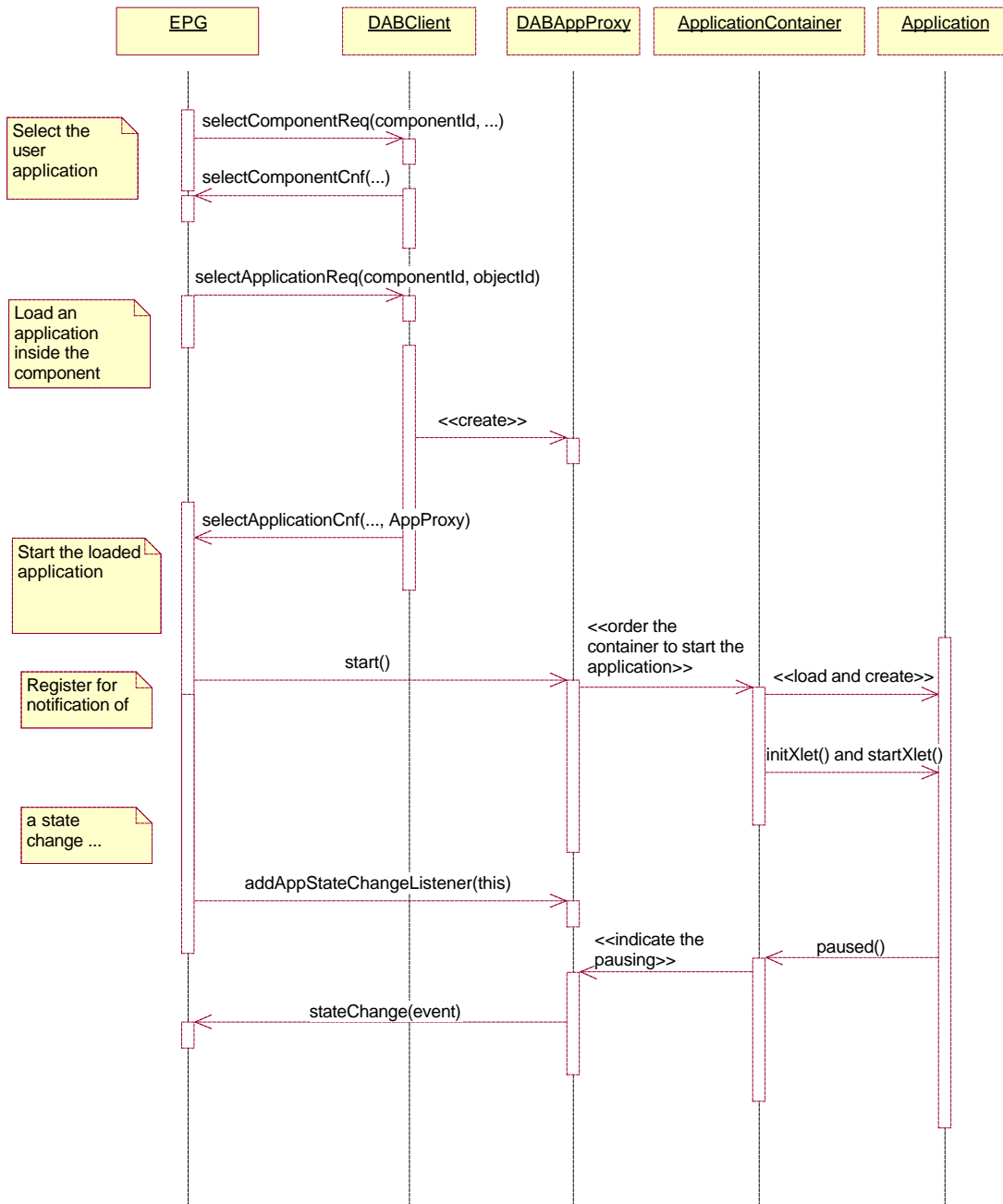


Figure 29: Launching an application

Example of selection of a Java object from an EPG application:

```

/*
 * selectJavaApplication

```

```

* select Java application from dab java runtime environment
* @param component          component owning the application
* @param applicationObject  object containing the application
*/
private void selectJavaApplication
(
    ComponentInfo component,
    ObjectId applicationObjectId
)
{
    try
    {
        ...
        /*
         * request the dab environment to select the application
         */
        dab.selectApplicationReq(component.getId(),applicationObjectId);
    }
    catch (DABException e)
    {
    }
    return;
}

```

...

the confirmation event carries the "Proxy" object for controlling the loaded application

```

/*
* selectApplicationCnf
* overload selectApplicationCnf method from DABAdapter
* handle received confirm event
* @param event          component owning the application
* @param applicationObject  object containing the application
*/
public void selectApplicationCnf(SelectApplicationCnfEvent event)
{
    if(event.getResult() == DABConstants.resultOK)
    {
        /*
         *Xlet container (implementation specific)
         */
        javaXletAppViewer = new XletAppViewer();

        /*
         * get application proxy from event
         */
        DABAppProxy javaAppProxy = event.getApplicationProxy();

        if(javaAppProxy == null)
        {
            return;
        }
        /*
         * pass application proxy to Xlet container
         */
        javaXletAppViewer.setXletAppProxy(javaAppProxy);
        /*
         * present application from the Xlet container
         */
        javaXletAppViewer.show();
    }
    return;
}

```

....

5.3 Security management

The DAB security model (see figure 30) that we present here is independent from the different security architectures presented in the different JDK releases (see Bibliography, "Implementing Protection Domains in the Java Development Kit 1.2" and "Java Security Architecture"). This means, the developer of the DAB VM may choose the security architecture that fits best to his/her own implementation requirements.

The security framework presented here is based on three main subsystems: policy profiling, resource access and security implementation subsystem.

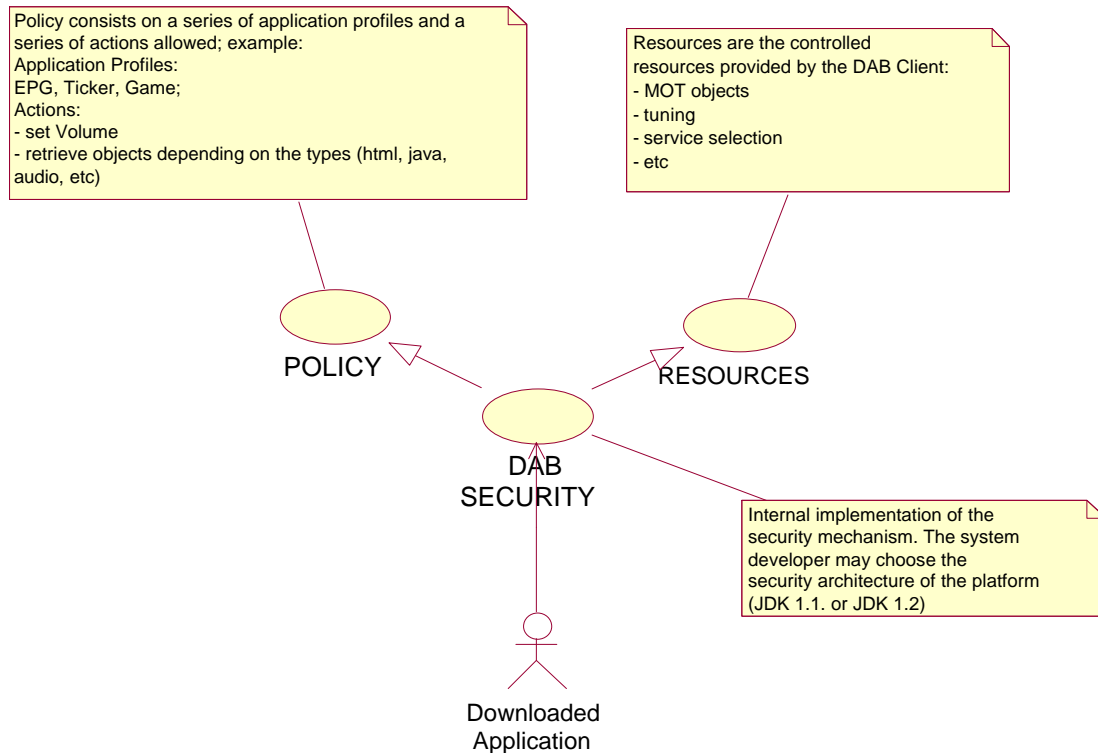


Figure 30: The DAB security model

The only access to DAB resources allowed to downloaded applications from the DAB channel is through the DABClient class; the security subsystem checks this accesses: the security system can return silently or throws a `Java.lang.SecurityException` (see clause 5.5 for other possible way to access system properties).

For simplifying the management of the security settings security profiles may be used. Three profiles are defined: EPG, MediaPresenter, and NODABAccess (see clause 6.3 for the coding). The EPG defines the class of applications that are allowed maximum access to the DAB terminal. The MediaPresenter profile refers to the applications that are allowed to access limited resources (for example only data file transmitted in the channel from where they are downloaded). Ultimately the NODABAccess profile: such profile can only use few terminal resources (Video, audio) but no DAB access is allowed.

In table 2 it is demonstrated, how these security profiles can be used to differentiate between the DAB commands considering the caller.

Table 2: Example of access to DAB resources vs. security profiles

Method name	EPG	MediaPresenter	NoDABAccess
getComponentInfoReq()	totally	no	no
setEnsembleInfoReq()	totally	no	no
getServiceInfoReq()	totally	partially	no
scanReq()	totally	no	no
searchReq()	totally	no	no
selectComponentReq()	totally	partially	no
selectObjectReq()	totally	partially	partially
selectReceptionInfoReq()	totally	partially	no
selectServiceInfoReq()	totally	partially	no
setVolumeReq()	totally	partially	partially
tuneReq()	totally	no	no
String System.getProperties(String)	totally	partially	no
String System.setProperties(String,String)	totally	no	no
selectApplicationReq()	totally	no	no
org.dvb.aplication.AppProxy methods	totally	no	no

Imagine the following scenario. An EPG application is downloaded from some channel (XXX Radio); the application can control the volume of the platform because is specified in the profile. The implementation of the YYY company provides a fine-grained security mechanism: the user can set min, max, normal volume for every different application profile and for every different content provider. The implementation of the ZZZ Company is more profile-oriented and it controls only the profile of the downloaded application. The internal mechanisms used in the two implementations are hidden to the loaded application.

For instance, we have the following code fragment in an application:

```

try
{
    dab.open();
}
catch(SecurityException e)
{
    // ...
}

try
{
    ab.setVolumeReq(10);
}
catch(SecurityException e)
{
    //...
}

```

In the DAB package it is implemented as follows:

```

public void open() throws DABException, SecurityException
{
    // perform a DAB security check using the AccessController
    AccessController.checkPermission(new DABRuntimePermission("open"));
    // the AC throws a securityException if the policy settings do not allow
    // the permission
    receiver = DABSystem.getReceiver();
    receiver.addReceiverListener(this);
}

```

5.4 Resource management

Resource management is used to share exclusive resources among different concurrently active clients. In the case of the DAB VM such kinds of resources are mostly related to the constraint set by the DAB receiver: e.g. only one ensemble can be accessed at the same time. We will present a model for the resource management and describe how resource conflicts are resolved within this model.

5.4.1 Model

The resource management is opaque. But resource conflicts are signalled. This means that the allocation of resources during a transaction is not visible outside the API. The VM has to care for all issues of resource management like allocation of resources, their release and deadlock avoidance. If the VM cannot allocate the required resources, the conflict resolution is initiated (see clause 5.4.2). If the conflict resolution fails, the failure of the transaction is indicated to the requesting client.

For the indication of failures because of resource conflicts a dedicated exception is used:

```
class ResourceConflictException extends DABException
{
ResourceConflictException() {}
ResourceConflictException(String message) {}
}
```

It is expected, when a request is issued (e.g. calling `selectReceptionInfoReq`) and the request fails due to resource conflicts, the `ResourceConflictException` is thrown.

5.4.2 Conflict Resolution

The VM handles the conflict resolution and negotiates with the involved parties. The conflict resolution consists of four turns (i.e. phases).

- 1) **Proceed:** Ask the requesting client for commitment despite resource conflicts.
- 2) **Probe:** If the client insists on his request, all other clients which own needed resources are asked whether they like to release their resources. If there is at least one who does not like to release a needed resource, the conflict resolution fails.
- 3) **Stop:** If all clients agreed to the request, they are actually asked to release the resources. When all clients have released the resources, the transaction is restarted.
- 4) **Preempt:** If there are some clients, which did not release the resources in the last turn, then these are preempted from the resources.

NOTE 1: In case a client does not respond to such requests, it is up to the VM to handle such behaviour. Typically, a timeout needs to be introduced and a default response has to be defined (e.g. it is assumed that the client agreed to release the resource). Like security management that should be configurable by the user.

NOTE 2: When clients are asked to release their resources (also for the willingness to do so), not the actual resources are specified but rather the related operation. That means the VM has to track the relationship between resources and operations.

The API supports the conflict resolution by the following notification:

```
interface DABListener
{
...
conflictResolutionNtf (ConflictResolutionNtfEvent event);
...
}

class ConflictResolutionNtfEvent
{
int getTransaction() {...}
int getTurn() {...}
}
```

```

    int getOperation() {...}
    int getSuboperation() {...}
}

class DABConstants
{
    ...
    public static final int conflictResolutionTurnProceed=0;
    public static final int conflictResolutionTurnProbe=1;
    public static final int conflictResolutionTurnStop=2;
    public static final int conflictResolutionTurnPreempt=3;

    public static final int conflictResolutionOperationNone=0;
    ... // plus constants for all

    // available operations (see DABSource)
}

```

The related event, ConflictResolutionNtfEvent, contains the information about the action:

- 1) getTransaction delivers the transaction number. This can be used to provide a transaction context.
- 2) getTurn returns a code for the turn of the resource conflict resolution protocol:
 - conflictResolutionTurnProceed:
This is sent to the client which requested the operation. It indicates that there is a resource conflict. The client is asked whether he likes to proceed.
 - conflictResolutionTurnProbe:
This notification is sent to all clients in order to probe for their willingness to release the needed resources.
 - conflictResolutionTurnStop:
The client is asked to stop the indicated operation in order to release the resources.
 - conflictResolutionTurnPreempt:
The client is informed that the indicated operation was stopped. This action shall normally only be taken, when the client failed to do a stop in the previous turn.
- 3) getOperation gives back a code of the involved operation (see extensions to DABConstants).
- 4) getSuboperation returns a code for the suboperation, which depends on the operation code. This is useful for "aggregated" operations like operationControl.

The response to the notifications described above are given using a request in DABSource:

```

interface DABSource
{
    ...
    respondConflictResolutionReq
    (
        int transaction,
        int turn,
        int operation,
        int suboperation,
        int answer
    );
    ...
}

class DABConstants

```



```

{
...
    public static final int conflictResolutionAnswerNo=0;
    public static final int conflictResolutionAnswerYes=1;
}

```

The argument `answer` refers to the answer with respect to the notification. The following values are possible:

- Turn Proceed
 - `conflictResolutionAnswerYes`: The client likes to have his request continued so that the negotiations for resources may start.
 - `conflictResolutionAnswerNo`: The client agrees to stop the request, which will result in a `ResourceConflictException` (see above).
- Turn Probe
 - `conflictResolutionAnswerYes`: The client is willing to stop the operation.
 - `conflictResolutionAnswerNo`: The client does not agree to stop the operation.
- Turn Stop
 - The request confirms that the operation was stopped by the client. No special value has to be specified for answer.

The other arguments of this request correspond to the attributes in the `ConflictResolutionNtfEvent`.

A confirmation will be sent for this request:

```

interface DABListener
{
    respondConflictResolutionCnf(conflictResolutionCnfEvent);
}

class conflictResolutionCnfEvent
{
    int getResult();
}

```

The method `getResult()` returns the result of the request.

The following example demonstrates these extensions. For this, we assume that we have selected an audio programme in a resident EPG and started a stock market ticker. Additionally, we have just launched a new EPG (which is provided by a broadcaster), which likes to do scanning.

In figure 31 it is indicated, how the emerging resource conflict is resolved. The VM signals the downloaded EPG the resource conflict (other applications have selected components inside the current ensemble).

The downloaded EPG decides to proceed with the conflict resolution. The VM then sends notifications to the resident EPG and to the ticker to probe for their agreement to stop the selections. When the agreement is given, they are actually asked to stop the selections. After that is done, the transaction for the scan is started and the `scanReq` call returns to the client.

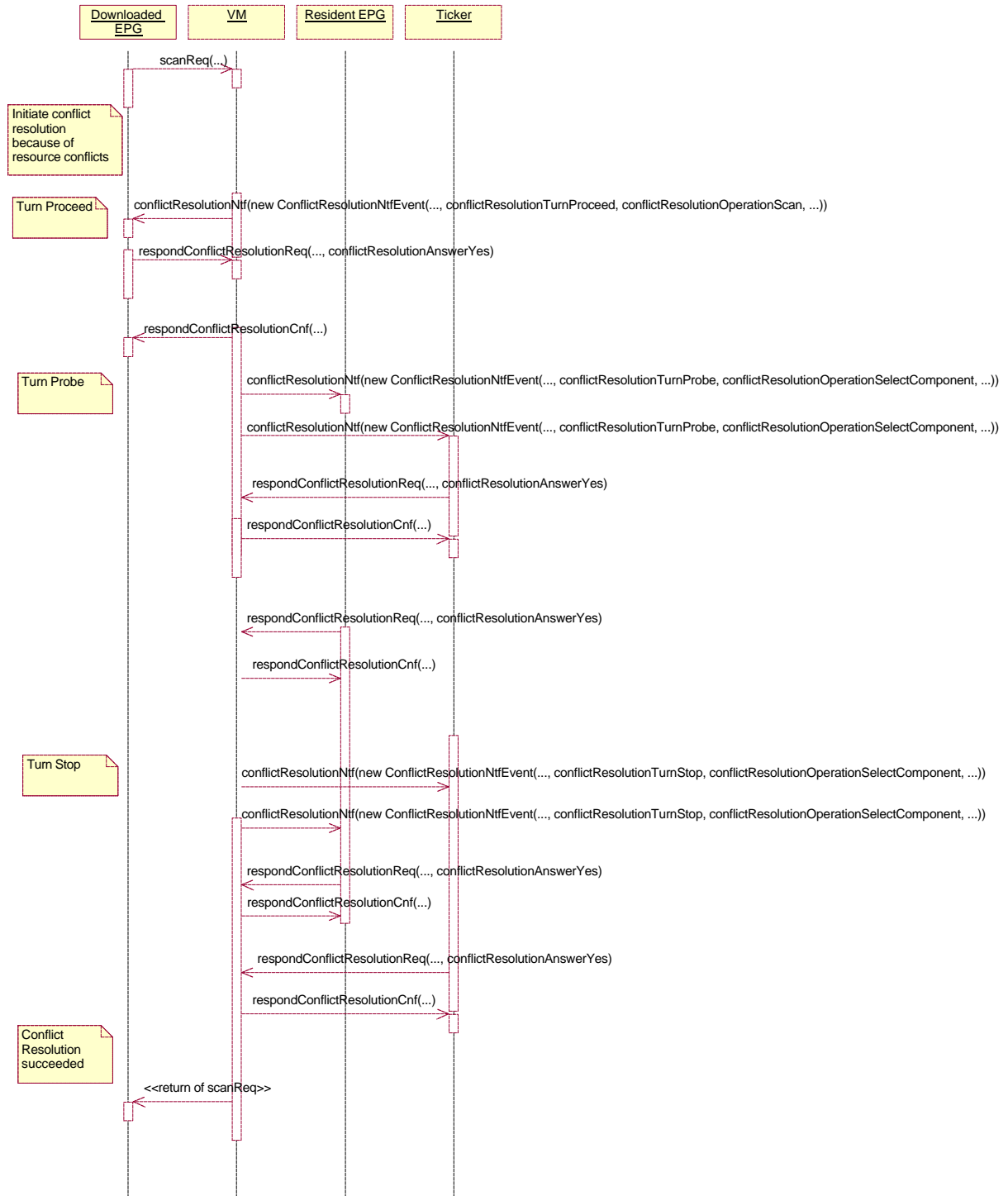


Figure 31: Example of conflict resolution

5.5 Configuration management

Configuration Management is related to the capability to profile the characteristics of the host terminal in a fine-grained manner.

Downloaded applications have to use a standard mechanism for determining the environment of the terminal for adapting their UI or their input routines. Additionally, the hosting environment has to give the application hooks to the runtime environment.

This leads to two classes of attributes, which the Xlet is interested in: platform scoped profile attributes and Xlet scoped profile attributes.

There is a very simple mechanism inside the Java runtime environment that permits to specify a set of properties for the Runtime system. Such properties are pairs of string values (key,value): for example:

- Java.home=c:\jdk1_2
- os.name=NT

These properties are set using methods located in the Java.lang.System class: System.setProperty(String key, String value) and String System.getProperty(String value).

dab.xlet.componentId	component identifier from which the Xlet was loaded (of type dab.si.ComponentId)
dab.xlet.serviceId	service identifier the Xlet belongs to. (of type dab.si.ServiceId)
dab.xlet.objectId	object identifier of the main archive of the Xlet (of type dab.ObjectId)
dab.xlet.panelContainer	reference to the GUI of the Xlet (of type Java.awt.Panel)
dab.xlet.datastream.in	standard input stream (of type Java.io.DataInputStream)
dab.xlet.datastream.out	standard output stream (of type Java.io.DataOutputStream)
dab.xlet.datastream.err	The standard error stream (of type Java.io.DataOutputStream)
dab.xlet.platform	The platform (of type Integer)
dab.xlet.accessType	The access type (of type Integer)
dab.xlet.content	The content type (of type Integer)
org.osgi.framework.BundleContext	A reference to a BundleContext object (an implementation of the interface org.osgi.framework.BundleContext) as specified by the Open Services Gateway initiative (OSGi), or null if the receiver does not support OSGi functions

Table 3: Predefined Xlet properties

For all the other attributes that are Xlet dependent we use the getXletProperties method of the XletContext. This method differs slightly from the System method: it returns objects (not strings). Using this method we can give to the Xlet objects needed for its runtime execution (see table: panelContainer, datastream.in, etc). In table 3, the predefined properties are listed.

The following example demonstrates the different attribute mechanisms:

Inside the application framework before activating the Xlet:

```
Panel xletPanel = new Panel();
appProxy.setAppProperty("dab.xlet.panelContainer", xletPanel);
appProxy.setAppProperty("dab.xlet.datastream.out", System.out);
```

Inside the Xlet:

```
public void initXlet(XletContext ctx) throws XletStateChangeException
{
    xPanel = (Java.awt.Panel)
        ctx.getXletProperty("dab.xlet.panelContainer");
    xOut = (Java.io.DataStream)
```

```

    ctx.getXletProperty("dab.xlet.datastream.out");
};

```

Access to OSGi functions

The Open Services Gateway initiative (OSGi) specifies an application framework which is used in driver information systems in order to allow for interaction between different applications. For DAB receivers which are part of a driver information and entertainment system the Xlet property "org.osgi.framework.BundleContext" facilitates that Xlets interact with other applications available in that system, e.g. GPS or a car navigation system.

If Xlets are executed in an OSGi based application framework they can get access to OSGi functions in the following way:

- Xlets which are designed to use OSGi functions must include an additional entry in the manifest according to the following format:

```
ImportPackage: <package 1>, ..., <package n>
```

where the expressions <package x> specify all Java packages which are required for the interaction with OSGi functions. The expression <package x> has the format

```
org.xyz.mypackage.mysubpackage; specification-version="1.2"
```

as defined in the OSGi specification.

- Xlets can call `XletContext.getXletProperty("org.osgi.framework.BundleContext")` in order to get an object which implements the interface `BundleContext` which is defined by the OSGi specification. This `BundleContext` object can be used to access OSGi functions as defined in the OSGi specification. All Java interfaces of OSGi functions which the Xlet is designed to access must be specified in the `ImportPackage` entries in the manifest as defined above.
- The method `getXletProperty("BundleContext")` may return null if OSGi is not supported by the platform or some of the packages specified by the Xlet in the `ImportPackage` entries in the manifest are not available. In this case the Xlet must be able to run without access to OSGi functions.

6 The User I/O Package

The DAB User IO package specifies the Java platform that should be supported for DABJava. It also defines the profiles for DABJava and the method for signalling the profile using the FIG0/13 User Application Type (see EN 300 401 [3]).

The DAB Java User IO package consists of the following parts:

- **Signalling.** This part defines the signalling in DAB of the DABJava profiles.
- **DABJava platform.** This part defines the current defined platforms and profiles that are used in DABJava devices. In the future other profiles can be added to the standard following the rules defined in clause 6.1.

6.1 Signalling

6.1.1 DAB Java User Application Profile (DJUAP)

DABJava is defined as a User Application (UA) type within FIG 0/13 (see [3],[4]). The DABJava UA parameters "Platform" and "Version" are used to signal the DABJava profile or application environment. They are carried in the User Application specific part of the FIG 0/13.

The User Application Data carries a three byte field. Byte 0 carries the Platform identifier, Byte 1 carries the Version identifier and Byte 2 carries the Access identifier.

The coding of the identifiers is described in the following clauses.

6.1.2 Platform

The DABJava UA parameter "Platform" signals the major version of the application environment (e.g. SPJP or NPJP). It should be a number between 0 and 255. New platform numbers must be registered at the WIRC by registration of a new DJUAP with the parameter "Version" set to 0.

DJUA platforms are not necessarily compatible with each other. This means that although platform A may partly cover platform B, an application for platform A will not necessarily run in the application environment of platform B.

6.1.3 Version

The UA parameter "Version" signals the minor version of the application environment. It should be a number between 0 and 255. New version numbers are obtained from the WIRC by updating an existing platform specification. The new version number shall be the previous version number increased by 1.

DABJava UA versions must be backwards compatible. That means that for a certain Platform where version A is older than version B, then an application written for version A must run in the application environment defined in version B.

6.1.4 Access

The UA parameter "Access" signals the access type. It should be a number between 0 and 255. New version numbers are obtained from the WIRC by updating an existing platform specification. The new version number shall be the previous version number increased by 1.

The following values are predefined:

- 0) EPG
- 1) MediaPresenter
- 2) NoDABAccess

6.1.5 Defined profiles

The currently defined profiles are described in details in clause 6.2, and they are signalled as follows:

6.1.5.1 Standard Personal Java Profile (SPJP)

The SPJP supports the APIs and the application environment of PersonalJava without the package "Java.net".

The UA parameters for this profile are:

Name	Value
Platform	0
Version	0
Access	0 to 2

6.1.6.2 Network enabled Personal Java Profile (NPJP)

The NPJP supports the APIs and the application environment of PersonalJava together with the class "org.dvb.net.DatagramSocketBufferControl" and the interface "Javax.tv.net.InterfaceMap" defined in the Java TV API Specification (see Bibliography).

The UA parameters for this profile are:

Name	Value
Platform	1
Version	0
Access	0 to 2

6.2 DABJava platforms

6.2.1 PersonalJava 1.1

Two profiles have been defined for DAB Java based on Personal Java. They modify the packages from PersonalJava 1.1 (see Bibliography) as shown in the following.

6.2.1.1 Core Packages

The packages used in PersonalJava 1.1 and common to the 2 profiles (6.16.1 and 6.1.6.2) are shown below, the supported packages are slightly modified for DAB Java profiles as described in clause 6.2.3:

- Java.applet
- Java.awt
- Java.awt.datatransfer
- Java.awt.event
- Java.awt.image
- Java.awt.peer
- Java.beans
- Java.io (*file support is optional*)
- Java.lang
- Java.lang.reflect
- Java.math (*package is optional*)
- Java.net (*some protocols are optional*)
- Java.rmi (*package is optional*)
- Java.rmi.dgc (*package is optional*)
- Java.rmi.registry (*package is optional*)
- Java.rmi.server (*package is optional*)
- Java.security (*package is optional*)
- Java.security.acl (*package is unsupported*)
- Java.security.interfaces (*package is optional*)
- Java.sql (*package is optional*)
- Java.text
- Java.text.resources (*modified and optional*)
- Java.util

- Java.util.zip

6.2.1.2 DABJava profiles: specific packages.

6.2.1.2.1 Standard Personal Java Profile (SPJP)

- Java.applet (package is optional)
- Java.beans (package is optional)
- Java.net (package is unsupported)

6.2.1.2.2 Network-enabled Personal Java Profile (NPJP)

- Java.applet (package is optional)
- Java.beans (package is optional)
- org.dvb.net.DatagramSocketBufferControl is added (see Bibliography, "DVB Java Specification")
- Javax.tv.net.InterfaceMap is added (see Bibliography, "DVB Java Specification")

Annex A (normative): The DAB Java classes

Package Summary		Page
dab		64
dab.data		154
dab.events		188
dab.si		236
dab.xlet		257

Package dab

Interface Summary		Page
AppStateChangeListener	AppStateChangeListener defines the interface for events originating from the DABAppProxy.	351
AppStateChangeEventListener	An interface that can be used to identify an application by its related controller.	351
DABAppProxy	This interface can be used to control applications that were launched using selectApplicationReq in DABSource.	340
DABListener	DABListener defines the interface for DAB listeners.	277
DABSource	DABSource defines the interface of a DAB resource (usually a DAB receiver).	262

Class Summary		Page
DABAdapter	The DABAdapter class provides default methods for the implementation of a DABListener.	345
DABClient	The DABClient class is used to access a DAB resource.	334
DABConstants	DABConstants contains the constants that are used inside the whole package (including the subpackages).	286
DABReceiverAddress	DABReceiverAddress is used to specify the location of DAB receivers.	275

Exception Summary		Page
DABConnectionException	The DABConnectionException is thrown when there are problems with the connection between the DAB client and the receiver.	333
DABException	DABException is the superclass for exceptions inside the DAB package.	285

DABNotAvailableException	The DABNotAvailableException is thrown when particular data is currently not available or even not at all available.	276
ResourceConflictException	The exception indicates unsolved resource conflicts.	261

Interface AppStateChangeListener

[dab](#)

public Interface **AppStateChangeListener**

AppStateChangeListener defines the interface for events originating from the DABAppProxy. Note: to support J2ME MIDP Profile AppStateChangeListener does not implement java.util.EventListener any more

Version:

0.3

See Also:

"Digital Video Broadcasting (DVB) Multimedia Home Platform (MHP), ETSI TS 101 812", [DABAppProxy](#)

Method Summary		Page
void	stateChange (dab.events.AppStateChangeEvent event)	351
	This method is used to signal a state change for the related application.	

Method Detail

stateChange

public void **stateChange**(dab.events.AppStateChangeEvent event)

This method is used to signal a state change for the related application.

Interface AppStateChangeEventSource

[dab](#)

All Known Subinterfaces:

[DABAppProxy](#)

public Interface **AppStateChangeEventSource**

An interface that can be used to identify an application by its related controller.

Version:

0.2

Class *DABAdapter*

[dab](#)

java.lang.Object

|

+--**dab.DABAdapter**

All Implemented Interfaces:

[DABListener](#)

public Class **DABAdapter**

extends java.lang.Object

implements [DABListener](#)

The *DABAdapter* class provides default methods for the implementation of a *DABListener*. The default behavior is that incoming events are ignored.

Version:

1.01

See Also:

[DABListener](#)

Constructor Summary	Page
DABAdapter ()	347

Method Summary	Page
void componentNtf (dab.events.ComponentNtfEvent e)	349
void conflictResolutionNtf (dab.events.ConflictResolutionNtfEvent e)	350
void drcModeNtf (dab.events.DRCModeNtfEvent e)	350
void getComponentInfoCnf (dab.events.GetComponentInfoCnfEvent e)	348
void getEnsembleInfoCnf (dab.events.GetEnsembleInfoCnfEvent e)	348
void getLocationInfoCnf (dab.events.GetLocationInfoCnfEvent e)	350
void getServiceInfoCnf (dab.events.GetServiceInfoCnfEvent e)	348
void locationInfoNtf (dab.events.LocationInfoNtfEvent e)	350
void objectNtf (dab.events.ObjectNtfEvent e)	349
void operationControlCnf (dab.events.OperationControlCnfEvent e)	350
void receptionInfoNtf (dab.events.ReceptionInfoNtfEvent e)	349
void respondConflictResolutionCnf (dab.events.RespondConflictResolutionCnfEvent e)	350
void scanCnf (dab.events.ScanCnfEvent e)	347
void scanNtf (dab.events.ScanNtfEvent e)	347
void searchCnf (dab.events.SearchCnfEvent e)	347
void searchNtf (dab.events.SearchNtfEvent e)	347
void selectApplicationCnf (dab.events.SelectApplicationCnfEvent e)	349
void selectComponentCnf (dab.events.SelectComponentCnfEvent e)	349

void	selectComponentStreamCnf (dab.events.SelectComponentStreamCnfEvent e)	349
void	selectObjectCnf (dab.events.SelectObjectCnfEvent e)	349
void	selectReceptionInfoCnf (dab.events.SelectReceptionInfoCnfEvent e)	348
void	selectSICnf (dab.events.SelectSICnfEvent e)	348
void	serviceFollowingNtf (dab.events.ServiceFollowingNtfEvent e)	350
void	siNtf (dab.events.SINtfEvent e)	348
void	systemFailureNtf (dab.events.SystemFailureNtfEvent e)	351
void	tuneCnf (dab.events.TuneCnfEvent e)	347

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface dab.[DABListener](#)

[componentNtf](#), [conflictResolutionNtf](#), [drcModeNtf](#), [getComponentInfoCnf](#), [getEnsembleInfoCnf](#), [getLocationInfoCnf](#), [getServiceInfoCnf](#), [locationInfoNtf](#), [objectNtf](#), [operationControlCnf](#), [receptionInfoNtf](#), [respondConflictResolutionCnf](#), [scanCnf](#), [scanNtf](#), [searchCnf](#), [searchNtf](#), [selectApplicationCnf](#), [selectComponentCnf](#), [selectComponentStreamCnf](#), [selectObjectCnf](#), [selectReceptionInfoCnf](#), [selectSICnf](#), [serviceFollowingNtf](#), [siNtf](#), [systemFailureNtf](#), [tuneCnf](#)

Constructor Detail

DABAdapter

public **DABAdapter**()

Method Detail

tuneCnf

public void **tuneCnf**(dab.events.TuneCnfEvent e)

Specified by:

[tuneCnf](#) in interface [DABListener](#)

searchCnf

public void **searchCnf**(dab.events.SearchCnfEvent e)

Specified by:

[searchCnf](#) in interface [DABListener](#)

searchNtf

public void **searchNtf**(dab.events.SearchNtfEvent e)

Specified by:

[searchNtf](#) in interface [DABListener](#)

scanCnf

```
public void scanCnf(dab.events.ScanCnfEvent e)
```

Specified by:

[scanCnf](#) in interface [DABListener](#)

scanNtf

```
public void scanNtf(dab.events.ScanNtfEvent e)
```

Specified by:

[scanNtf](#) in interface [DABListener](#)

selectSICnf

```
public void selectSICnf(dab.events.SelectSICnfEvent e)
```

Specified by:

[selectSICnf](#) in interface [DABListener](#)

siNtf

```
public void siNtf(dab.events.SINtfEvent e)
```

Specified by:

[siNtf](#) in interface [DABListener](#)

getEnsembleInfoCnf

```
public void getEnsembleInfoCnf(dab.events.GetEnsembleInfoCnfEvent e)
```

Specified by:

[getEnsembleInfoCnf](#) in interface [DABListener](#)

getServiceInfoCnf

```
public void getServiceInfoCnf(dab.events.GetServiceInfoCnfEvent e)
```

Specified by:

[getServiceInfoCnf](#) in interface [DABListener](#)

getComponentInfoCnf

```
public void getComponentInfoCnf(dab.events.GetComponentInfoCnfEvent e)
```

Specified by:

[getComponentInfoCnf](#) in interface [DABListener](#)

selectReceptionInfoCnf

```
public void selectReceptionInfoCnf(dab.events.SelectReceptionInfoCnfEvent e)
```

Specified by:

[selectReceptionInfoCnf](#) in interface [DABListener](#)

receptionInfoNtf

```
public void receptionInfoNtf(dab.events.ReceptionInfoNtfEvent e)
```

Specified by:

[receptionInfoNtf](#) in interface [DABListener](#)

selectComponentCnf

```
public void selectComponentCnf(dab.events.SelectComponentCnfEvent e)
```

Specified by:

[selectComponentCnf](#) in interface [DABListener](#)

selectComponentStreamCnf

```
public void selectComponentStreamCnf(dab.events.SelectComponentStreamCnfEvent e)
```

Specified by:

[selectComponentStreamCnf](#) in interface [DABListener](#)

componentNtf

```
public void componentNtf(dab.events.ComponentNtfEvent e)
```

Specified by:

[componentNtf](#) in interface [DABListener](#)

selectObjectCnf

```
public void selectObjectCnf(dab.events.SelectObjectCnfEvent e)
```

Specified by:

[selectObjectCnf](#) in interface [DABListener](#)

selectApplicationCnf

```
public void selectApplicationCnf(dab.events.SelectApplicationCnfEvent e)
```

Specified by:

[selectApplicationCnf](#) in interface [DABListener](#)

objectNtf

```
public void objectNtf(dab.events.ObjectNtfEvent e)
```

Specified by:

[objectNtf](#) in interface [DABListener](#)

getLocationInfoCnf

```
public void getLocationInfoCnf(dab.events.GetLocationInfoCnfEvent e)
```

Specified by:

[getLocationInfoCnf](#) in interface [DABListener](#)

locationInfoNtf

```
public void locationInfoNtf(dab.events.LocationInfoNtfEvent e)
```

Specified by:

[locationInfoNtf](#) in interface [DABListener](#)

conflictResolutionNtf

```
public void conflictResolutionNtf(dab.events.ConflictResolutionNtfEvent e)
```

Specified by:

[conflictResolutionNtf](#) in interface [DABListener](#)

respondConflictResolutionCnf

```
public void respondConflictResolutionCnf(dab.events.RespondConflictResolutionCnfEvent e)
```

Specified by:

[respondConflictResolutionCnf](#) in interface [DABListener](#)

operationControlCnf

```
public void operationControlCnf(dab.events.OperationControlCnfEvent e)
```

Specified by:

[operationControlCnf](#) in interface [DABListener](#)

serviceFollowingNtf

```
public void serviceFollowingNtf(dab.events.ServiceFollowingNtfEvent e)
```

Specified by:

[serviceFollowingNtf](#) in interface [DABListener](#)

drcModeNtf

```
public void drcModeNtf(dab.events.DRCModeNtfEvent e)
```

Specified by:

[drcModeNtf](#) in interface [DABListener](#)

systemFailureNtf

```
public void systemFailureNtf(dab.events.SystemFailureNtfEvent e)
```

Specified by:

[systemFailureNtf](#) in interface [DABListener](#)

Interface DABAppProxy

[dab](#)

All Superinterfaces:

[AppStateChangeEventSource](#)

```
public Interface DABAppProxy
```

```
extends AppStateChangeEventSource
```

This interface can be used to control applications that were launched using `selectApplicationReq` in `DABSource`.

Version:

0.2

See Also:

"Digital Video Broadcasting (DVB) Multimedia Home Platform (MHP), ETSI TS 101 812"

Field Summary		Page
int	DESTROYED the final state of the application - no further actions are possible	342
int	INITED the application is loaded and initialized, but there is no activity yet	342
int	LOADED the application is loaded	342

int	<u>NOT_LOADED</u> the application was selected, but is not yet loaded	342
int	<u>PAUSED</u> the application is paused, which means it is not active	342
int	<u>STARTED</u> the application is active	342

Method Summary		Page
void	<u>addAppStateChangeListener</u> (dab.AppStateChangeListener listener) adds a listener for application state changes	343
java.lang.Object	<u>getAppProperty</u> (java.lang.String key) gets a property of the application.	345
int	<u>getState</u> () returns the current state of the application (cf. the defined constants)	343
void	<u>init</u> () initialises the application.	343
void	<u>load</u> () loads the classes of the application.	343
void	<u>pause</u> () pauses the application.	344
void	<u>removeAppStateChangeListener</u> (dab.AppStateChangeListener listener) removes a listener for application state changes	343
void	<u>resume</u> () resumes the application.	344
void	<u>setAppProperty</u> (java.lang.String key, java.lang.Object value) sets a property of the application.	345
void	<u>start</u> () starts the application.	344
void	<u>stop</u> (boolean forced) requests to stop the application.	344

Field Detail

DESTROYED

```
public static final int DESTROYED
```

the final state of the application - no further actions are possible

NOT_LOADED

```
public static final int NOT_LOADED
```

the application was selected, but is not yet loaded

LOADED

```
public static final int LOADED
```

the application is loaded

INITED

```
public static final int INITED
```

the application is loaded and initialized, but there is no activity yet

PAUSED

```
public static final int PAUSED
```

the application is paused, which means it is not active

STARTED

```
public static final int STARTED
```

the application is active

Method Detail**addAppStateChangeListener**

```
public void addAppStateChangeListener(dab.AppStateChangeListener listener)
```

adds a listener for application state changes

removeAppStateChangeListener

```
public void removeAppStateChangeListener(dab.AppStateChangeListener listener)
```

removes a listener for application state changes

getState

```
public int getState()  
    throws java.lang.SecurityException
```

returns the current state of the application (cf. the defined constants)

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to retrieve the application state

load

```
public void load()  
    throws java.lang.SecurityException
```

loads the classes of the application. The state of the application changes to LOADED. This action is only successful, if the application was not loaded before. A state change event is signalled in any case.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to load the application

init

```
public void init()  
    throws java.lang.SecurityException
```

initialises the application. The routine `initXlet` in the related application will be called. The application is afterwards in the INITED state.

This action is only successful, if the application was not initialised before. If the application was not loaded, the application will first be loaded and then initialised.

A state change event is signalled in any case. An additional state change is signalled if the application also has to be loaded. In this case first the state change to LOADED is signalled and afterwards that one to INITED.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to initialise the application

See Also:

[Xlet](#)

pause

```
public void pause()  
    throws java.lang.SecurityException
```

pauses the application. The routine `pauseXlet` in the related application will be called. The application is afterwards in the PAUSED state.

This action is only successful, if the application is either in the INITED state or in the STARTED state.

A state change event is signalled in any case.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to pause the application

See Also:

[Xlet](#)

resume

```
public void resume()  
    throws java.lang.SecurityException
```

resumes the application. The routine `startXlet` in the related application will be called. The application is afterwards in the STARTED state.

This action is only successful, if the application was in the PAUSED state.

A state change event is signalled in any case.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to resume the application

See Also:

[Xlet](#)

start

```
public void start()  
    throws java.lang.SecurityException
```

starts the application. The routine `startXlet` in the related application will be called. The application is afterwards in the `STARTED` state.

This action is only successful, if the application was not paused or destroyed. If the application was not loaded, the application will first be loaded, then initialised and finally be started. If the application was not initialised, the application will be initialised and then started.

A state change event is signalled in any case. Additional state changes will also be signalled (e.g. `NOT_LOADED - LOADED - INITED - STARTED`).

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to start the application

See Also:

[Xlet](#)

stop

```
public void stop(boolean forced)  
    throws java.lang.SecurityException
```

requests to stop the application. The routine `destroyXlet` in the related application will be called. The application is afterwards in the `DESTROYED` state.

This action is only successful, if the application was not destroyed.

A state change event is signalled in any case.

Parameters:

`forced` - if set to true the application is asked to stop and may refuse. if set to false, the application is stopped in any case.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to stop the application

See Also:

[Xlet](#)

setAppProperty

```
public void setAppProperty(java.lang.String key,  
                           java.lang.Object value)  
    throws java.lang.SecurityException
```

sets a property of the application.

Parameters:

`key` - the name of the property

`value` - the new value of the property

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to set an property of the application

getAppProperty

```
public java.lang.Object getAppProperty(java.lang.String key)
                               throws java.lang.SecurityException
```

gets a property of the application. It returns the value of the property or NULL if the property is not defined.

Parameters:

`key` - the name of the property

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to retrieve an property of the application

Class DABClient[dab](#)

```
java.lang.Object
|
+--dab.DABClient
```

All Implemented Interfaces:

[DABSource](#)

```
public Class DABClient
```

```
extends java.lang.Object
```

```
implements DABSource
```

The DABClient class is used to access a DAB resource. Usually the DAB resource might be a receiver that resides on the same host or is at least directly connected to it. But, it could also be a network device.

Note: the actual interface is defined in DABSource.

Version:

1.05

See Also:

[DABSource](#)

Constructor Summary		Page
DABClient ()	Create a DABClient object	336

Method Summary		Page
synchronized void	addDABListener (dab.DABListener listener) Register a DAB listener.	336
void	close () The connection to the current receiver is closed.	337
void	getComponentInfoReq (dab.si.ComponentId id)	338
void	getEnsembleInfoReq (dab.si.EnsembleId id)	338

void	getLocationInfoReq (int type, int mode, int desiredDelta, int desiredAccuracy)	340
void	getServiceInfoReq (dab.si.ServiceId id)	338
void	open () A connection to the default receiver is opened.	336
void	open (dab.DABReceiverAddress receiverAddress) A connection to the given receiver is opened.	337
void	operationControlReq (int attribute, java.lang.Object value)	340
synchronized void	removeDABListener (dab.DABListener listener) Removes the given listener from the list of DAB Listeners.	336
void	respondConflictResolutionReq (int transaction, int turn, int operation, int suboperation, int answer)	340
void	scanReq (int searchMode, int tables, int startFrequency, int stopFrequency, int transmissionModes, int notifications)	338
void	searchReq (int searchMode, int tables, int startFrequency, int stopFrequency, int transmissionModes, int notifications)	337
void	selectApplicationReq (dab.si.ComponentId componentId, dab.data.ObjectId objectId)	339
void	selectComponentReq (dab.si.ComponentId id, int selectionMode)	339
void	selectComponentStreamReq (dab.si.ComponentId componentId)	339
void	selectObjectReq (dab.si.ComponentId id, dab.data.ObjectId objectId, int requestMode, boolean replaceSelections, int deliveryMode, int cacheHint)	339
void	selectReceptionInfoReq (boolean synchronizationNotification, boolean bitErrorRateNotifications, boolean muteStateNotifications, boolean requestOnce)	339
void	selectSIReq (boolean ensembleInfo, boolean serviceInfo, boolean componentInfo, boolean autoDelivery)	338
void	tuneReq (int tuneFrequency, int transmissionMode)	337

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface dab.[DABSource](#)

[getComponentInfoReq](#), [getEnsembleInfoReq](#), [getLocationInfoReq](#), [getServiceInfoReq](#), [operationControlReq](#), [respondConflictResolutionReq](#), [scanReq](#), [searchReq](#), [selectApplicationReq](#), [selectComponentReq](#), [selectComponentStreamReq](#), [selectObjectReq](#), [selectReceptionInfoReq](#), [selectSIReq](#), [tuneReq](#)

Constructor Detail

DABClient

```
public DABClient()
```

Create a DABClient object

Method Detail**addDABListener**

```
public synchronized void addDABListener(dab.DABListener listener)
```

Register a DAB listener. DAB events, that relate to this client, are distributed to all registered listeners.

See Also:

[DABEvent](#), [DABListener](#), [removeDABListener](#)

removeDABListener

```
public synchronized void removeDABListener(dab.DABListener listener)
```

Removes the given listener from the list of DAB Listeners.

See Also:

[DABListener](#), [addDABListener](#)

open

```
public void open()  
    throws DABException,  
           java.lang.SecurityException
```

A connection to the default receiver is opened.

Throws:

[DABException](#) - when the client could not be registered

[SecurityException](#) - when the application controlling the *DABClient* does not have the permission to call the open method

open

```
public void open(dab.DABReceiverAddress receiverAddress)  
    throws DABException,  
           java.lang.SecurityException
```

A connection to the given receiver is opened. This method is only supported in configurations with multiple receivers.

Parameters:

`receiverAddress` - This parameter specifies the address of the receiver to be used

Throws:

[DABException](#) - when the client could not be registered

[SecurityException](#) - when the application controlling the *DABClient* does not have the permission to call the open method

close

```
public void close()  
    throws DABException
```

The connection to the current receiver is closed. All ongoing transactions of the client are canceled.

Throws:

[DABException](#) - when no connection was opened

tuneReq

```
public void tuneReq(int tuneFrequency,  
                    int transmissionMode)  
    throws DABException,  
           java.lang.SecurityException
```

Specified by:

[tuneReq](#) in interface [DABSource](#)

searchReq

```
public void searchReq(int searchMode,  
                       int tables,  
                       int startFrequency,  
                       int stopFrequency,  
                       int transmissionModes,  
                       int notifications)  
    throws DABException,  
           java.lang.SecurityException
```

Specified by:

[searchReq](#) in interface [DABSource](#)

scanReq

```
public void scanReq(int searchMode,  
                    int tables,  
                    int startFrequency,  
                    int stopFrequency,  
                    int transmissionModes,  
                    int notifications)  
    throws DABException,  
           java.lang.SecurityException
```

Specified by:

[scanReq](#) in interface [DABSource](#)

selectSReq

```
public void selectSReq(boolean ensembleInfo,  
                       boolean serviceInfo,  
                       boolean componentInfo,  
                       boolean autoDelivery)  
    throws DABException,  
           java.lang.SecurityException
```

Specified by:

[selectSReq](#) in interface [DABSource](#)

getEnsembleInfoReq

```
public void getEnsembleInfoReq(dab.si.EnsembleId id)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[getEnsembleInfoReq](#) in interface [DABSource](#)

getServiceInfoReq

```
public void getServiceInfoReq(dab.si.ServiceId id)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[getServiceInfoReq](#) in interface [DABSource](#)

getComponentInfoReq

```
public void getComponentInfoReq(dab.si.ComponentId id)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[getComponentInfoReq](#) in interface [DABSource](#)

selectReceptionInfoReq

```
public void selectReceptionInfoReq(boolean synchronizationNotification,
    boolean bitErrorRateNotifications,
    boolean muteStateNotifications,
    boolean requestOnce)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[selectReceptionInfoReq](#) in interface [DABSource](#)

selectComponentReq

```
public void selectComponentReq(dab.si.ComponentId id,
    int selectionMode)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[selectComponentReq](#) in interface [DABSource](#)

selectComponentStreamReq

```
public void selectComponentStreamReq(dab.si.ComponentId componentId)
    throws DABException,
    java.lang.SecurityException
```

Specified by:

[selectComponentStreamReq](#) in interface [DABSource](#)

selectApplicationReq

```
public void selectApplicationReq(dab.si.ComponentId componentId,
    dab.data.ObjectId objectId)
    throws DABException,
    java.lang.SecurityException
```

Specified by:

[selectApplicationReq](#) in interface [DABSource](#)

selectObjectReq

```
public void selectObjectReq(dab.si.ComponentId id,
    dab.data.ObjectId objectId,
    int requestMode,
    boolean replaceSelections,
    int deliveryMode,
    int cacheHint)
    throws DABException,
    java.lang.SecurityException
```

Specified by:

[selectObjectReq](#) in interface [DABSource](#)

getLocationInfoReq

```
public void getLocationInfoReq(int type,
    int mode,
    int desiredDelta,
    int desiredAccuracy)
    throws DABException,
    java.lang.SecurityException
```

Specified by:

[getLocationInfoReq](#) in interface [DABSource](#)

respondConflictResolutionReq

```
public void respondConflictResolutionReq(int transaction,
    int turn,
    int operation,
    int suboperation,
    int answer)
```

Specified by:

[respondConflictResolutionReq](#) in interface [DABSource](#)

operationControlReq

```
public void operationControlReq(int attribute,
                               java.lang.Object value)
                               throws DABException,
                               java.lang.SecurityException
```

Specified by:

[operationControlReq](#) in interface [DABSource](#)

Class **DABConnectionException**

[dab](#)

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--dab.DABException
            |
            +--dab.DABConnectionException
```

All Implemented Interfaces:

java.io.Serializable

public Class **DABConnectionException**

extends [DABException](#)

The `DABConnectionException` is thrown when there are problems with the connection between the DAB client and the receiver.

Version:

1.0

Constructor Summary	Page
DABConnectionException()	334

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail**DABConnectionException**public **DABConnectionException**()**Class DABConstants**[dab](#)

java.lang.Object

|

+--**dab.DABConstants**public Class **DABConstants**

extends java.lang.Object

DABConstants contains the constants that are used inside the whole package (including the subpackages).

If you would like to add any new constants, please contact WorldDAB Information and Registration Centre, Wyvil Court, Wyvil Road, LONDON SW8 2TG, England, Tel: +44 171 896 90 51, Fax: +44 171 896 90 55, E-mail: worlddab-irc@worlddab.org.

Version:

1.07

Field Summary		Page
static final int	acsEuroCryptEN50094	316
static final int	acsNone	316
static final int	acsNR_MSK	316
static final int	acsReserverd1	316
static final int	acsReserverd2	316
static final int	acsReserverd3	316
static final int	acsReserverd4	316
static final int	acsReserverd5	316
static final int	announcementAlarm	309
static final int	announcementAreaWeatherFlash	310

static final int	<u>announcementEventAnnouncement</u>	310
static final int	<u>announcementNewsFlash</u>	310
static final int	<u>announcementReserved1</u>	310
static final int	<u>announcementReserved2</u>	310
static final int	<u>announcementReserved3</u>	310
static final int	<u>announcementReserved4</u>	310
static final int	<u>announcementReserved5</u>	310
static final int	<u>announcementReserved6</u>	310
static final int	<u>announcementReserved7</u>	310
static final int	<u>announcementReserved8</u>	310
static final int	<u>announcementRoadTrafficFlash</u>	309
static final int	<u>announcementSpecialEvent</u>	310
static final int	<u>announcementTransportFlash</u>	309
static final int	<u>announcementWarning_Service</u>	309
static final int	<u>bitErrorRateLevel1</u>	307
static final int	<u>bitErrorRateLevel2</u>	307
static final int	<u>bitErrorRateLevel3</u>	307
static final int	<u>bitErrorRateLevel4</u>	307
static final int	<u>bitErrorRateLevel5</u>	307
static final int	<u>bitErrorRateLevelUnknown</u>	307
static final int	<u>charsetCompleteEBULatin</u>	328
static final int	<u>charsetEBUArabic_HebrewETC</u>	328
static final int	<u>charsetEBUCyrillicGreek</u>	328
static final int	<u>charsetISOLatinAlphabetNo2</u>	328
static final int	<u>componentTypeBackgroundSound</u>	317
static final int	<u>componentTypeBroadcastWebSite</u>	317
static final int	<u>componentTypeDynamicLabel</u>	317
static final int	<u>componentTypeEmergencyWarningSystem</u>	317
static final int	<u>componentTypeForegroundSound</u>	316
static final int	<u>componentTypeInteractiveTextTransmissionSystem</u>	317
static final int	<u>componentTypeIPTunneling</u>	317
static final int	<u>componentTypeJava</u>	317
static final int	<u>componentTypeMultichannelAudio</u>	317
static final int	<u>componentTypePaging</u>	317

static final int	<u>componentTypeSlideshow</u>	317
static final int	<u>componentTypeTrafficMessageChannel</u>	317
static final int	<u>componentTypeUnspecified</u>	316
static final int	<u>conflictResolutionAnswerNo</u>	333
static final int	<u>conflictResolutionAnswerYes</u>	333
static final int	<u>conflictResolutionOperationGetComponentInfoReq</u>	332
static final int	<u>conflictResolutionOperationGetEnsembleInfoReq</u>	332
static final int	<u>conflictResolutionOperationGetLocationInfoReq</u>	332
static final int	<u>conflictResolutionOperationGetServiceInfoReq</u>	332
static final int	<u>conflictResolutionOperationNone</u>	331
static final int	<u>conflictResolutionOperationOperationControlReq</u>	332
static final int	<u>conflictResolutionOperationScanReq</u>	332
static final int	<u>conflictResolutionOperationSearchReq</u>	331
static final int	<u>conflictResolutionOperationSelectApplicationReq</u>	332
static final int	<u>conflictResolutionOperationSelectComponentReq</u>	332
static final int	<u>conflictResolutionOperationSelectComponentStreamReq</u>	332
static final int	<u>conflictResolutionOperationSelectObjectReq</u>	332
static final int	<u>conflictResolutionOperationSelectReceptionInfoReq</u>	332
static final int	<u>conflictResolutionOperationSelectSIReq</u>	332
static final int	<u>conflictResolutionOperationTuneReq</u>	331
static final int	<u>conflictResolutionSuboperationNone</u>	333
static final int	<u>conflictResolutionTurnPreempt</u>	331
static final int	<u>conflictResolutionTurnProbe</u>	331
static final int	<u>conflictResolutionTurnProceed</u>	331
static final int	<u>conflictResolutionTurnStop</u>	331
static final int	<u>countryAlbania</u>	311
static final int	<u>countryAlgeria</u>	311
static final int	<u>countryAndorra</u>	311
static final int	<u>countryAustria</u>	311
static final int	<u>countryAzores_Portugal</u>	311
static final int	<u>countryBelarus</u>	311
static final int	<u>countryBelgium</u>	311
static final int	<u>countryBosniaHerzegovina</u>	311
static final int	<u>countryBulgaria</u>	311

static final int	<u>countryCanaries_Spain</u>	311
static final int	<u>countryCroatia</u>	311
static final int	<u>countryCyprus</u>	311
static final int	<u>countryCzechRepublic</u>	312
static final int	<u>countryDenmark</u>	312
static final int	<u>countryEgypt</u>	312
static final int	<u>countryEstonia</u>	312
static final int	<u>countryFaroe_Denmark</u>	312
static final int	<u>countryFinland</u>	312
static final int	<u>countryFrance</u>	312
static final int	<u>countryGermany1</u>	312
static final int	<u>countryGermany2</u>	312
static final int	<u>countryGibraltar_UnitedKingdom</u>	312
static final int	<u>countryGreece</u>	312
static final int	<u>countryHungary</u>	312
static final int	<u>countryIceland</u>	313
static final int	<u>countryIraq</u>	313
static final int	<u>countryIreland</u>	313
static final int	<u>countryIsrael</u>	313
static final int	<u>countryItaly</u>	313
static final int	<u>countryJordan</u>	313
static final int	<u>countryLatvia</u>	313
static final int	<u>countryLebanon</u>	313
static final int	<u>countryLibya</u>	313
static final int	<u>countryLiechtenstein</u>	313
static final int	<u>countryLithuania</u>	313
static final int	<u>countryLuxembourg</u>	313
static final int	<u>countryMacedonia</u>	314
static final int	<u>countryMadeira_Portugal</u>	314
static final int	<u>countryMalta</u>	314
static final int	<u>countryMoldova</u>	314
static final int	<u>countryMonaco</u>	314
static final int	<u>countryMorocco</u>	314
static final int	<u>countryNetherlands</u>	314

static final int	<u>countryNorways</u>	314
static final int	<u>countryPalestine</u>	314
static final int	<u>countryPoland</u>	314
static final int	<u>countryPortugal</u>	314
static final int	<u>countryRomania</u>	314
static final int	<u>countryRussianFederation</u>	315
static final int	<u>countrySanMarino</u>	315
static final int	<u>countrySlovakia</u>	315
static final int	<u>countrySlovenia</u>	315
static final int	<u>countrySpain</u>	315
static final int	<u>countrySweden</u>	315
static final int	<u>countrySwitzerland</u>	315
static final int	<u>countrySyrianArabRepublic</u>	315
static final int	<u>countryTunisia</u>	315
static final int	<u>countryTurkey</u>	315
static final int	<u>countryUkraine</u>	315
static final int	<u>countryUnitedKingdom</u>	315
static final int	<u>countryVaticanCityState</u>	316
static final int	<u>countryYugoslavia</u>	316
static final int	<u>deliveryModeComplete</u>	304
static final int	<u>deliveryModePartial</u>	304
static final int	<u>language2C</u>	321
static final int	<u>language2D</u>	321
static final int	<u>language2E</u>	321
static final int	<u>language2F</u>	321
static final int	<u>language41</u>	328
static final int	<u>language42</u>	328
static final int	<u>language43</u>	328
static final int	<u>language44</u>	328
static final int	<u>languageAlbanian</u>	318
static final int	<u>languageAmharic</u>	323
static final int	<u>languageArabic</u>	323
static final int	<u>languageArmenian</u>	323
static final int	<u>languageAssamese</u>	323

static final int	<u>languageAzerbaijani</u>	323
static final int	<u>languageBackgroundSoundCleanFeed</u>	328
static final int	<u>languageBambora</u>	323
static final int	<u>languageBasque</u>	319
static final int	<u>languageBelorussian</u>	323
static final int	<u>languageBengali</u>	323
static final int	<u>languageBreton</u>	318
static final int	<u>languageBulgarian</u>	323
static final int	<u>languageBurmese</u>	324
static final int	<u>languageCatalan</u>	318
static final int	<u>languageChinese</u>	324
static final int	<u>languageChurash</u>	324
static final int	<u>languageCroatian</u>	318
static final int	<u>languageCzech</u>	318
static final int	<u>languageDanish</u>	318
static final int	<u>languageDari</u>	324
static final int	<u>languageDutch</u>	320
static final int	<u>languageEnglish</u>	318
static final int	<u>languageEsperanto</u>	318
static final int	<u>languageEstonian</u>	318
static final int	<u>languageFaroese</u>	319
static final int	<u>languageFinnish</u>	321
static final int	<u>languageFlemish</u>	321
static final int	<u>languageFrench</u>	319
static final int	<u>languageFrisian</u>	319
static final int	<u>languageFulani</u>	324
static final int	<u>languageGaelic</u>	319
static final int	<u>languageGalician</u>	319
static final int	<u>languageGeorgian</u>	324
static final int	<u>languageGerman</u>	318
static final int	<u>languageGreek</u>	324
static final int	<u>languageGujurati</u>	324
static final int	<u>languageGurani</u>	324
static final int	<u>languageHausa</u>	324

static final int	<u>languageHebrew</u>	324
static final int	<u>languageHindi</u>	324
static final int	<u>languageHungarian</u>	320
static final int	<u>languageIcelandic</u>	319
static final int	<u>languageIndonesian</u>	325
static final int	<u>languageIrish</u>	319
static final int	<u>languageItalian</u>	319
static final int	<u>languageJapanese</u>	325
static final int	<u>languageKannada</u>	325
static final int	<u>languageKazakh</u>	325
static final int	<u>languageKhmer</u>	325
static final int	<u>languageKorean</u>	325
static final int	<u>languageLaotian</u>	325
static final int	<u>languageLappish</u>	319
static final int	<u>languageLatin</u>	319
static final int	<u>languageLatvian</u>	319
static final int	<u>languageLithuanian</u>	320
static final int	<u>languageLuxembourgian</u>	320
static final int	<u>languageMacedonian</u>	325
static final int	<u>languageMalagasay</u>	325
static final int	<u>languageMalaysian</u>	325
static final int	<u>languageMaltese</u>	320
static final int	<u>languageMarathi</u>	325
static final int	<u>languageMoldavian</u>	325
static final int	<u>languageNational30</u>	321
static final int	<u>languageNational31</u>	322
static final int	<u>languageNational32</u>	322
static final int	<u>languageNational33</u>	322
static final int	<u>languageNational34</u>	322
static final int	<u>languageNational35</u>	322
static final int	<u>languageNational36</u>	322
static final int	<u>languageNational37</u>	322
static final int	<u>languageNational38</u>	322
static final int	<u>languageNational39</u>	322

static final int	<u>languageNational3A</u>	322
static final int	<u>languageNational3B</u>	322
static final int	<u>languageNational3C</u>	322
static final int	<u>languageNational3D</u>	323
static final int	<u>languageNational3E</u>	323
static final int	<u>languageNational3F</u>	323
static final int	<u>languageNdebele</u>	326
static final int	<u>languageNepali</u>	326
static final int	<u>languageNorwegian</u>	320
static final int	<u>languageOccitan</u>	320
static final int	<u>languageOriya</u>	326
static final int	<u>languagePapamiento</u>	326
static final int	<u>languagePersian</u>	326
static final int	<u>languagePolish</u>	320
static final int	<u>languagePortuguese</u>	320
static final int	<u>languagePunjabi</u>	326
static final int	<u>languagePushtu</u>	326
static final int	<u>languageQuechua</u>	326
static final int	<u>languageRomanian</u>	320
static final int	<u>languageRomansh</u>	320
static final int	<u>languageRussian</u>	326
static final int	<u>languageRuthenian</u>	326
static final int	<u>languageSerbian</u>	320
static final int	<u>languageSerbo_Croat</u>	326
static final int	<u>languageShona</u>	326
static final int	<u>languageSinhalese</u>	327
static final int	<u>languageSlovak</u>	321
static final int	<u>languageSlovene</u>	321
static final int	<u>languageSomali</u>	327
static final int	<u>languageSpanish</u>	318
static final int	<u>languageSranan_Tongo</u>	327
static final int	<u>languageSwahili</u>	327
static final int	<u>languageSwedish</u>	321
static final int	<u>languageTadzhik</u>	327

static final int	<u>languageTamil</u>	327
static final int	<u>languageTatar</u>	327
static final int	<u>languageTelugu</u>	327
static final int	<u>languageThai</u>	327
static final int	<u>languageTurkish</u>	321
static final int	<u>languageUkrainian</u>	327
static final int	<u>languageUnkown</u>	317
static final int	<u>languageUrdu</u>	327
static final int	<u>languageUzbek</u>	327
static final int	<u>languageVietnamese</u>	328
static final int	<u>languageWalloon</u>	321
static final int	<u>languageWelsh</u>	318
static final int	<u>languageZulu</u>	328
static final int	<u>locationInfoOnce</u>	329
static final int	<u>locationInfoPeriodByDistance</u>	329
static final int	<u>locationInfoPeriodByTime</u>	329
static final int	<u>locationInfoPosition</u>	329
static final int	<u>locationInfoRegionId</u>	329
static final int	<u>locationInfoStop</u>	329
static final int	<u>muteStateMuting</u>	307
static final int	<u>muteStateNotMuting</u>	307
static final int	<u>muteStatePartialMuting</u>	307
static final int	<u>muteStateUnknown</u>	307
static final int	<u>notification16kHzStep</u>	302
static final int	<u>notificationComponentAdded</u>	303
static final int	<u>notificationComponentChanged</u>	303
static final int	<u>notificationComponentRemoved</u>	303
static final int	<u>notificationEnsembleAdded</u>	302
static final int	<u>notificationEnsembleChanged</u>	303
static final int	<u>notificationEnsembleFound</u>	302
static final int	<u>notificationEnsembleRemoved</u>	302
static final int	<u>notificationFrequencyStep</u>	302
static final int	<u>notificationNone</u>	302
static final int	<u>notificationOff</u>	302

static final int	<u>notificationSearchStarted</u>	302
static final int	<u>notificationServiceAdded</u>	303
static final int	<u>notificationServiceChanged</u>	303
static final int	<u>notificationServiceRemoved</u>	303
static final int	<u>notificationTableEntry</u>	302
static final int	<u>operationControlGetDRCMode</u>	330
static final int	<u>operationControlGetDRCModeNotifications</u>	330
static final int	<u>operationControlGetServiceFollowing</u>	330
static final int	<u>operationControlGetServiceFollowingNotifications</u>	330
static final int	<u>operationControlGetVolume</u>	329
static final int	<u>operationControlSetDRCMode</u>	330
static final int	<u>operationControlSetServiceFollowing</u>	329
static final int	<u>operationControlSetVolume</u>	329
static final int	<u>requestModeOff</u>	303
static final int	<u>requestModeOnce</u>	304
static final int	<u>requestModeUpdate</u>	304
static final int	<u>resultApplicationNotFound</u> the requested/indicated application was not found	300
static final int	<u>resultComponentNotFound</u> the requested/indicated component was not found	300
static final int	<u>resultEnsembleNotFound</u> the requested/indicated ensemble was not found	300
static final int	<u>resultFatalError</u> a system error occurred (either related to hardware or to the operating system)	299
static final int	<u>resultInternalError</u> an internal error occurred in the DAB VM (e.g. an implementation error)	299
static final int	<u>resultInvalidParameter</u> the value of some parameter is not correct	299
static final int	<u>resultNonApplicableFunction</u> the operation is not applicable in the current context	300
static final int	<u>resultNotSupported</u> the requested operation is not supported	299
static final int	<u>resultObjectNotSelected</u> the indicated object was not be selected (in advance)	300
static final int	<u>resultOK</u> no problems occurred	299

static final int	<u>resultOutOfMemory</u> the system ran out of memory	300
static final int	<u>resultServiceNotFound</u> the requested/indicated service was not found	300
static final int	<u>searchCanadaFrequencyTableLBand</u>	302
static final int	<u>searchCEPTFrequencyTableBandIII</u>	302
static final int	<u>searchCEPTFrequencyTableLBand</u>	302
static final int	<u>searchModel6kHzSteps</u>	301
static final int	<u>searchModeAutomatic</u>	301
static final int	<u>searchModeContinuous</u>	301
static final int	<u>searchModeDown</u>	301
static final int	<u>searchModeUp</u>	301
static final int	<u>searchModeUseFrequencyRange</u>	301
static final int	<u>searchModeUseTables</u>	301
static final int	<u>selectionModeAdd</u>	303
static final int	<u>selectionModeRemove</u>	303
static final int	<u>selectionModeRemoveAll</u>	303
static final int	<u>selectionModeReplace</u>	303
static final int	<u>selectionStateDelayed</u>	308
static final int	<u>selectionStateOk</u>	307
static final int	<u>selectionStateTerminated</u>	308
static final int	<u>serviceElementTypeComponent</u>	329
static final int	<u>serviceElementTypeEnsemble</u>	329
static final int	<u>serviceElementTypeService</u>	329
static final int	<u>serviceElementTypeUndefined</u>	328
static final int	<u>serviceFollowingLeavingService</u>	330
static final int	<u>serviceFollowingSelectingService</u>	330
static final int	<u>serviceFollowingTryingAlternativeService</u>	330
static final int	<u>serviceSelectorAnnouncement</u>	309
static final int	<u>serviceSelectorAnnouncementSupport</u>	309
static final int	<u>serviceSelectorCountry</u>	308
static final int	<u>serviceSelectorDate</u>	308
static final int	<u>serviceSelectorDynamicProgrammeType</u>	308
static final int	<u>serviceSelectorFrequency</u>	308
static final int	<u>serviceSelectorLabel</u>	308

static final int	<u>serviceSelectorLanguage</u>	309
static final int	<u>serviceSelectorNone</u>	308
static final int	<u>serviceSelectorRegion</u>	308
static final int	<u>serviceSelectorRegionId</u>	309
static final int	<u>serviceSelectorRegionLabel</u>	309
static final int	<u>serviceSelectorStartObject</u>	309
static final int	<u>serviceSelectorStaticProgrammeType</u>	308
static final int	<u>serviceSelectorTime</u>	308
static final int	<u>serviceSelectorTimeOffset</u>	308
static final int	<u>serviceTypeAudioService</u>	309
static final int	<u>serviceTypeDataService</u>	309
static final int	<u>streamTypeAudio</u>	330
static final int	<u>streamTypeFIDC</u>	331
static final int	<u>streamTypePacket</u>	330
static final int	<u>streamTypeStream</u>	330
static final int	<u>streamTypeXPAD</u>	330
static final int	<u>subscriberInfoExpiredSubscription</u>	331
static final int	<u>subscriberInfoNoAlgorithm</u>	331
static final int	<u>subscriberInfoNoCA</u>	331
static final int	<u>subscriberInfoNoSubscription</u>	331
static final int	<u>syncStateDABSignalDetected</u>	304
static final int	<u>syncStateFICReadable</u>	304
static final int	<u>syncStateNotSynchronized</u>	304
static final int	<u>syncStateSynchronizationStateUnknown</u>	304
static final int	<u>syncStateTimeAndFrequencySynchronized</u>	304
static final int	<u>syncUpdateBitErrorRateState</u>	306
static final int	<u>syncUpdateMuteState</u>	307
static final int	<u>syncUpdateSynchronizationState</u>	306
static final int	<u>transmissionMode1</u>	301
static final int	<u>transmissionMode2</u>	301
static final int	<u>transmissionMode3</u>	301
static final int	<u>transmissionMode4</u>	301
static final int	<u>transmissionModeAutomatic</u>	300
static final int	<u>transmissionModeUnknown</u>	301

static final int	tuneStateNotTuned	304
static final int	tuneStateSearching	305
static final int	tuneStateTuned	305
static final int	tuneStateTuning	304
static final int	tuneStateUnknown	304
static final int	updatedAnnouncement	306
static final int	updatedAnnouncementSupport	306
static final int	updatedAudioComponent	306
static final int	updatedBitrate	306
static final int	updatedCountry	305
static final int	updatedDate	305
static final int	updatedDynamicProgrammeType	305
static final int	updatedFrequency	305
static final int	updatedLabel	305
static final int	updatedLanguage	306
static final int	updatedNone	305
static final int	updatedObjectDirectory	306
static final int	updatedProgrammeNumber	306
static final int	updatedRegion	305
static final int	updatedRegionId	306
static final int	updatedRegionLabel	306
static final int	updatedStartObject	306
static final int	updatedStaticProgrammeType	305
static final int	updatedTime	305
static final int	updatedTimeOffset	305

Constructor Summary		Page
	DABConstants ()	333

Method Summary		Page
static java.lang.String	result2String (int result) The method returns a string for the given result code.	333

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Field Detail**resultOK**

```
public static final int resultOK
```

no problems occurred

resultNotSupported

```
public static final int resultNotSupported
```

the requested operation is not supported

resultFatalError

```
public static final int resultFatalError
```

a system error occurred (either related to hardware or to the operating system)

resultInternalError

```
public static final int resultInternalError
```

an internal error occurred in the DAB VM (e.g. an implementation error)

resultInvalidParameter

```
public static final int resultInvalidParameter
```

the value of some parameter is not correct

resultOutOfMemory

```
public static final int resultOutOfMemory
```

the system ran out of memory

resultNonApplicableFunction

```
public static final int resultNonApplicableFunction
```

the operation is not applicable in the current context

resultEnsembleNotFound

```
public static final int resultEnsembleNotFound
```

the requested/indicated ensemble was not found

resultServiceNotFound

```
public static final int resultServiceNotFound
```

the requested/indicated service was not found

resultComponentNotFound

```
public static final int resultComponentNotFound
```

the requested/indicated component was not found

resultObjectNotSelected

```
public static final int resultObjectNotSelected
```

the indicated object was not be selected (in advance)

resultApplicationNotFound

```
public static final int resultApplicationNotFound
```

the requested/indicated application was not found

transmissionModeAutomatic

```
public static final int transmissionModeAutomatic
```

transmissionMode1

```
public static final int transmissionMode1
```

transmissionMode2

```
public static final int transmissionMode2
```

transmissionMode3

```
public static final int transmissionMode3
```

transmissionMode4

```
public static final int transmissionMode4
```

transmissionModeUnknown

```
public static final int transmissionModeUnknown
```

searchModeAutomatic

```
public static final int searchModeAutomatic
```

searchMode16kHzSteps

```
public static final int searchMode16kHzSteps
```

searchModeUp

```
public static final int searchModeUp
```

searchModeDown

```
public static final int searchModeDown
```

searchModeUseTables

```
public static final int searchModeUseTables
```

searchModeUseFrequencyRange

```
public static final int searchModeUseFrequencyRange
```

searchModeContinuous

```
public static final int searchModeContinuous
```

searchCEPTFrequencyTableBandIII

```
public static final int searchCEPTFrequencyTableBandIII
```

searchCEPTFrequencyTableLBand

```
public static final int searchCEPTFrequencyTableLBand
```

searchCanadaFrequencyTableLBand

```
public static final int searchCanadaFrequencyTableLBand
```

notificationOff

```
public static final int notificationOff
```

notificationFrequencyStep

```
public static final int notificationFrequencyStep
```

notification16kHzStep

```
public static final int notification16kHzStep
```

notificationTableEntry

```
public static final int notificationTableEntry
```

notificationEnsembleFound

```
public static final int notificationEnsembleFound
```

notificationSearchStarted

```
public static final int notificationSearchStarted
```

notificationNone

```
public static final int notificationNone
```

notificationEnsembleAdded

```
public static final int notificationEnsembleAdded
```

notificationEnsembleRemoved

```
public static final int notificationEnsembleRemoved
```

notificationEnsembleChanged

```
public static final int notificationEnsembleChanged
```

notificationServiceAdded

```
public static final int notificationServiceAdded
```

notificationServiceRemoved

```
public static final int notificationServiceRemoved
```

notificationServiceChanged

```
public static final int notificationServiceChanged
```

notificationComponentAdded

```
public static final int notificationComponentAdded
```

notificationComponentRemoved

```
public static final int notificationComponentRemoved
```

notificationComponentChanged

```
public static final int notificationComponentChanged
```

selectionModeReplace

```
public static final int selectionModeReplace
```

selectionModeAdd

```
public static final int selectionModeAdd
```

selectionModeRemove

```
public static final int selectionModeRemove
```

selectionModeRemoveAll

```
public static final int selectionModeRemoveAll
```

requestModeOff

```
public static final int requestModeOff
```

requestModeOnce

```
public static final int requestModeOnce
```

requestModeUpdate

```
public static final int requestModeUpdate
```

deliveryModeComplete

```
public static final int deliveryModeComplete
```

deliveryModePartial

```
public static final int deliveryModePartial
```

syncStateSynchronizationStateUnknown

```
public static final int syncStateSynchronizationStateUnknown
```

syncStateNotSynchronized

```
public static final int syncStateNotSynchronized
```

syncStateDABSignalDetected

```
public static final int syncStateDABSignalDetected
```

syncStateTimeAndFrequencySynchronized

```
public static final int syncStateTimeAndFrequencySynchronized
```

syncStateFICReadable

```
public static final int syncStateFICReadable
```

tuneStateUnknown

```
public static final int tuneStateUnknown
```

tuneStateNotTuned

```
public static final int tuneStateNotTuned
```

tuneStateTuning

```
public static final int tuneStateTuning
```

tuneStateSearching

```
public static final int tuneStateSearching
```

tuneStateTuned

```
public static final int tuneStateTuned
```

updatedNone

```
public static final int updatedNone
```

updatedLabel

```
public static final int updatedLabel
```

updatedCountry

```
public static final int updatedCountry
```

updatedFrequency

```
public static final int updatedFrequency
```

updatedDate

```
public static final int updatedDate
```

updatedTime

```
public static final int updatedTime
```

updatedTimeOffset

```
public static final int updatedTimeOffset
```

updatedRegion

```
public static final int updatedRegion
```

updatedStaticProgrammeType

```
public static final int updatedStaticProgrammeType
```

updatedDynamicProgrammeType

```
public static final int updatedDynamicProgrammeType
```

updatedAnnouncement

```
public static final int updatedAnnouncement
```

updatedLanguage

```
public static final int updatedLanguage
```

updatedRegionId

```
public static final int updatedRegionId
```

updatedRegionLabel

```
public static final int updatedRegionLabel
```

updatedAnnouncementSupport

```
public static final int updatedAnnouncementSupport
```

updatedStartObject

```
public static final int updatedStartObject
```

updatedObjectDirectory

```
public static final int updatedObjectDirectory
```

updatedProgrammeNumber

```
public static final int updatedProgrammeNumber
```

updatedAudioComponent

```
public static final int updatedAudioComponent
```

updatedBitrate

```
public static final int updatedBitrate
```

syncUpdateSynchronizationState

```
public static final int syncUpdateSynchronizationState
```

syncUpdateBitErrorRateState

```
public static final int syncUpdateBitErrorRateState
```

syncUpdateMuteState

```
public static final int syncUpdateMuteState
```

bitErrorRateLevelUnknown

```
public static final int bitErrorRateLevelUnknown
```

bitErrorRateLevel1

```
public static final int bitErrorRateLevel1
```

bitErrorRateLevel2

```
public static final int bitErrorRateLevel2
```

bitErrorRateLevel3

```
public static final int bitErrorRateLevel3
```

bitErrorRateLevel4

```
public static final int bitErrorRateLevel4
```

bitErrorRateLevel5

```
public static final int bitErrorRateLevel5
```

muteStateUnknown

```
public static final int muteStateUnknown
```

muteStateMuting

```
public static final int muteStateMuting
```

muteStatePartialMuting

```
public static final int muteStatePartialMuting
```

muteStateNotMuting

```
public static final int muteStateNotMuting
```

selectionStateOk

```
public static final int selectionStateOk
```

selectionStateDelayed

```
public static final int selectionStateDelayed
```

selectionStateTerminated

```
public static final int selectionStateTerminated
```

serviceSelectorNone

```
public static final int serviceSelectorNone
```

serviceSelectorLabel

```
public static final int serviceSelectorLabel
```

serviceSelectorCountry

```
public static final int serviceSelectorCountry
```

serviceSelectorFrequency

```
public static final int serviceSelectorFrequency
```

serviceSelectorDate

```
public static final int serviceSelectorDate
```

serviceSelectorTime

```
public static final int serviceSelectorTime
```

serviceSelectorTimeOffset

```
public static final int serviceSelectorTimeOffset
```

serviceSelectorRegion

```
public static final int serviceSelectorRegion
```

serviceSelectorStaticProgrammeType

```
public static final int serviceSelectorStaticProgrammeType
```

serviceSelectorDynamicProgrammeType

```
public static final int serviceSelectorDynamicProgrammeType
```

serviceSelectorAnnouncement

```
public static final int serviceSelectorAnnouncement
```

serviceSelectorLanguage

```
public static final int serviceSelectorLanguage
```

serviceSelectorRegionId

```
public static final int serviceSelectorRegionId
```

serviceSelectorRegionLabel

```
public static final int serviceSelectorRegionLabel
```

serviceSelectorAnnouncementSupport

```
public static final int serviceSelectorAnnouncementSupport
```

serviceSelectorStartObject

```
public static final int serviceSelectorStartObject
```

serviceTypeAudioService

```
public static final int serviceTypeAudioService
```

serviceTypeDataService

```
public static final int serviceTypeDataService
```

announcementAlarm

```
public static final int announcementAlarm
```

announcementRoadTrafficFlash

```
public static final int announcementRoadTrafficFlash
```

announcementTransportFlash

```
public static final int announcementTransportFlash
```

announcementWarning_Service

```
public static final int announcementWarning_Service
```

announcementNewsFlash

```
public static final int announcementNewsFlash
```

announcementAreaWeatherFlash

```
public static final int announcementAreaWeatherFlash
```

announcementEventAnnouncement

```
public static final int announcementEventAnnouncement
```

announcementSpecialEvent

```
public static final int announcementSpecialEvent
```

announcementReserved1

```
public static final int announcementReserved1
```

announcementReserved2

```
public static final int announcementReserved2
```

announcementReserved3

```
public static final int announcementReserved3
```

announcementReserved4

```
public static final int announcementReserved4
```

announcementReserved5

```
public static final int announcementReserved5
```

announcementReserved6

```
public static final int announcementReserved6
```

announcementReserved7

```
public static final int announcementReserved7
```

announcementReserved8

```
public static final int announcementReserved8
```

countryAlbania

```
public static final int countryAlbania
```

countryAlgeria

```
public static final int countryAlgeria
```

countryAndorra

```
public static final int countryAndorra
```

countryAustria

```
public static final int countryAustria
```

countryAzores_Portugal

```
public static final int countryAzores_Portugal
```

countryBelgium

```
public static final int countryBelgium
```

countryBelarus

```
public static final int countryBelarus
```

countryBosniaHerzegovina

```
public static final int countryBosniaHerzegovina
```

countryBulgaria

```
public static final int countryBulgaria
```

countryCanaries_Spain

```
public static final int countryCanaries_Spain
```

countryCroatia

```
public static final int countryCroatia
```

countryCyprus

```
public static final int countryCyprus
```

countryCzechRepublic

```
public static final int countryCzechRepublic
```

countryDenmark

```
public static final int countryDenmark
```

countryEgypt

```
public static final int countryEgypt
```

countryEstonia

```
public static final int countryEstonia
```

countryFaroe_Denmark

```
public static final int countryFaroe_Denmark
```

countryFinland

```
public static final int countryFinland
```

countryFrance

```
public static final int countryFrance
```

countryGermany1

```
public static final int countryGermany1
```

countryGermany2

```
public static final int countryGermany2
```

countryGibraltar_UnitedKingdom

```
public static final int countryGibraltar_UnitedKingdom
```

countryGreece

```
public static final int countryGreece
```

countryHungary

```
public static final int countryHungary
```

countryIceland

```
public static final int countryIceland
```

countryIraq

```
public static final int countryIraq
```

countryIreland

```
public static final int countryIreland
```

countryIsrael

```
public static final int countryIsrael
```

countryItaly

```
public static final int countryItaly
```

countryJordan

```
public static final int countryJordan
```

countryLatvia

```
public static final int countryLatvia
```

countryLebanon

```
public static final int countryLebanon
```

countryLibya

```
public static final int countryLibya
```

countryLiechtenstein

```
public static final int countryLiechtenstein
```

countryLithuania

```
public static final int countryLithuania
```

countryLuxembourg

```
public static final int countryLuxembourg
```

countryMacedonia

```
public static final int countryMacedonia
```

countryMadeira_Portugal

```
public static final int countryMadeira_Portugal
```

countryMalta

```
public static final int countryMalta
```

countryMoldova

```
public static final int countryMoldova
```

countryMonaco

```
public static final int countryMonaco
```

countryMorocco

```
public static final int countryMorocco
```

countryNetherlands

```
public static final int countryNetherlands
```

countryNorways

```
public static final int countryNorways
```

countryPalestine

```
public static final int countryPalestine
```

countryPoland

```
public static final int countryPoland
```

countryPortugal

```
public static final int countryPortugal
```

countryRomania

```
public static final int countryRomania
```

countryRussianFederation

```
public static final int countryRussianFederation
```

countrySanMarino

```
public static final int countrySanMarino
```

countrySlovakia

```
public static final int countrySlovakia
```

countrySlovenia

```
public static final int countrySlovenia
```

countrySpain

```
public static final int countrySpain
```

countrySweden

```
public static final int countrySweden
```

countrySwitzerland

```
public static final int countrySwitzerland
```

countrySyrianArabRepublic

```
public static final int countrySyrianArabRepublic
```

countryTunisia

```
public static final int countryTunisia
```

countryTurkey

```
public static final int countryTurkey
```

countryUkraine

```
public static final int countryUkraine
```

countryUnitedKingdom

```
public static final int countryUnitedKingdom
```

countryVaticanCityState

```
public static final int countryVaticanCityState
```

countryYugoslavia

```
public static final int countryYugoslavia
```

acsNone

```
public static final int acsNone
```

acsNR_MSK

```
public static final int acsNR_MSK
```

acsEuroCryptEN50094

```
public static final int acsEuroCryptEN50094
```

acsReserverd1

```
public static final int acsReserverd1
```

acsReserverd2

```
public static final int acsReserverd2
```

acsReserverd3

```
public static final int acsReserverd3
```

acsReserverd4

```
public static final int acsReserverd4
```

acsReserverd5

```
public static final int acsReserverd5
```

componentTypeUnspecified

```
public static final int componentTypeUnspecified
```

componentTypeForegroundSound

```
public static final int componentTypeForegroundSound
```

componentTypeBackgroundSound

```
public static final int componentTypeBackgroundSound
```

componentTypeMultichannelAudio

```
public static final int componentTypeMultichannelAudio
```

componentTypeTrafficMessageChannel

```
public static final int componentTypeTrafficMessageChannel
```

componentTypeEmergencyWarningSystem

```
public static final int componentTypeEmergencyWarningSystem
```

componentTypeInteractiveTextTransmissionSystem

```
public static final int componentTypeInteractiveTextTransmissionSystem
```

componentTypePaging

```
public static final int componentTypePaging
```

componentTypeDynamicLabel

```
public static final int componentTypeDynamicLabel
```

componentTypeSlideshow

```
public static final int componentTypeSlideshow
```

componentTypeBroadcastWebSite

```
public static final int componentTypeBroadcastWebSite
```

componentTypeJava

```
public static final int componentTypeJava
```

componentTypeIPTunneling

```
public static final int componentTypeIPTunneling
```

languageUnkown

```
public static final int languageUnkown
```

languageAlbanian

```
public static final int languageAlbanian
```

languageBreton

```
public static final int languageBreton
```

languageCatalan

```
public static final int languageCatalan
```

languageCroatian

```
public static final int languageCroatian
```

languageWelsh

```
public static final int languageWelsh
```

languageCzech

```
public static final int languageCzech
```

languageDanish

```
public static final int languageDanish
```

languageGerman

```
public static final int languageGerman
```

languageEnglish

```
public static final int languageEnglish
```

languageSpanish

```
public static final int languageSpanish
```

languageEsperanto

```
public static final int languageEsperanto
```

languageEstonian

```
public static final int languageEstonian
```

languageBasque

```
public static final int languageBasque
```

languageFaroese

```
public static final int languageFaroese
```

languageFrench

```
public static final int languageFrench
```

languageFrisian

```
public static final int languageFrisian
```

languageIrish

```
public static final int languageIrish
```

languageGaelic

```
public static final int languageGaelic
```

languageGalician

```
public static final int languageGalician
```

languageIcelandic

```
public static final int languageIcelandic
```

languageItalian

```
public static final int languageItalian
```

languageLappish

```
public static final int languageLappish
```

languageLatin

```
public static final int languageLatin
```

languageLatvian

```
public static final int languageLatvian
```

languageLuxembourgian

```
public static final int languageLuxembourgian
```

languageLithuanian

```
public static final int languageLithuanian
```

languageHungarian

```
public static final int languageHungarian
```

languageMaltese

```
public static final int languageMaltese
```

languageDutch

```
public static final int languageDutch
```

languageNorwegian

```
public static final int languageNorwegian
```

languageOccitan

```
public static final int languageOccitan
```

languagePolish

```
public static final int languagePolish
```

languagePortuguese

```
public static final int languagePortuguese
```

languageRomanian

```
public static final int languageRomanian
```

languageRomansh

```
public static final int languageRomansh
```

languageSerbian

```
public static final int languageSerbian
```

languageSlovak

```
public static final int languageSlovak
```

languageSlovene

```
public static final int languageSlovene
```

languageFinnish

```
public static final int languageFinnish
```

languageSwedish

```
public static final int languageSwedish
```

languageTurkish

```
public static final int languageTurkish
```

languageFlemish

```
public static final int languageFlemish
```

languageWalloon

```
public static final int languageWalloon
```

language2C

```
public static final int language2C
```

language2D

```
public static final int language2D
```

language2E

```
public static final int language2E
```

language2F

```
public static final int language2F
```

languageNational30

```
public static final int languageNational30
```

languageNational31

```
public static final int languageNational31
```

languageNational32

```
public static final int languageNational32
```

languageNational33

```
public static final int languageNational33
```

languageNational34

```
public static final int languageNational34
```

languageNational35

```
public static final int languageNational35
```

languageNational36

```
public static final int languageNational36
```

languageNational37

```
public static final int languageNational37
```

languageNational38

```
public static final int languageNational38
```

languageNational39

```
public static final int languageNational39
```

languageNational3A

```
public static final int languageNational3A
```

languageNational3B

```
public static final int languageNational3B
```

languageNational3C

```
public static final int languageNational3C
```

languageNational3D

```
public static final int languageNational3D
```

languageNational3E

```
public static final int languageNational3E
```

languageNational3F

```
public static final int languageNational3F
```

languageAmharic

```
public static final int languageAmharic
```

languageArabic

```
public static final int languageArabic
```

languageArmenian

```
public static final int languageArmenian
```

languageAssamese

```
public static final int languageAssamese
```

languageAzerbaijani

```
public static final int languageAzerbaijani
```

languageBambora

```
public static final int languageBambora
```

languageBelorussian

```
public static final int languageBelorussian
```

languageBengali

```
public static final int languageBengali
```

languageBulgarian

```
public static final int languageBulgarian
```

languageBurmese

```
public static final int languageBurmese
```

languageChinese

```
public static final int languageChinese
```

languageChurash

```
public static final int languageChurash
```

languageDari

```
public static final int languageDari
```

languageFulani

```
public static final int languageFulani
```

languageGeorgian

```
public static final int languageGeorgian
```

languageGreek

```
public static final int languageGreek
```

languageGujurati

```
public static final int languageGujurati
```

languageGurani

```
public static final int languageGurani
```

languageHausa

```
public static final int languageHausa
```

languageHebrew

```
public static final int languageHebrew
```

languageHindi

```
public static final int languageHindi
```

languageIndonesian

```
public static final int languageIndonesian
```

languageJapanese

```
public static final int languageJapanese
```

languageKannada

```
public static final int languageKannada
```

languageKazakh

```
public static final int languageKazakh
```

languageKhmer

```
public static final int languageKhmer
```

languageKorean

```
public static final int languageKorean
```

languageLaotian

```
public static final int languageLaotian
```

languageMacedonian

```
public static final int languageMacedonian
```

languageMalagasay

```
public static final int languageMalagasay
```

languageMalaysian

```
public static final int languageMalaysian
```

languageMoldavian

```
public static final int languageMoldavian
```

languageMarathi

```
public static final int languageMarathi
```

languageNdebele

```
public static final int languageNdebele
```

languageNepali

```
public static final int languageNepali
```

languageOriya

```
public static final int languageOriya
```

languagePapamiento

```
public static final int languagePapamiento
```

languagePersian

```
public static final int languagePersian
```

languagePunjabi

```
public static final int languagePunjabi
```

languagePushtu

```
public static final int languagePushtu
```

languageQuechua

```
public static final int languageQuechua
```

languageRussian

```
public static final int languageRussian
```

languageRuthenian

```
public static final int languageRuthenian
```

languageSerbo_Croat

```
public static final int languageSerbo_Croat
```

languageShona

```
public static final int languageShona
```

languageSinhalese

```
public static final int languageSinhalese
```

languageSomali

```
public static final int languageSomali
```

languageSranan_Tongo

```
public static final int languageSranan_Tongo
```

languageSwahili

```
public static final int languageSwahili
```

languageTadzhik

```
public static final int languageTadzhik
```

languageTamil

```
public static final int languageTamil
```

languageTatar

```
public static final int languageTatar
```

languageTelugu

```
public static final int languageTelugu
```

languageThai

```
public static final int languageThai
```

languageUkrainian

```
public static final int languageUkrainian
```

languageUrdu

```
public static final int languageUrdu
```

languageUzbek

```
public static final int languageUzbek
```

languageVietnamese

```
public static final int languageVietnamese
```

languageZulu

```
public static final int languageZulu
```

language44

```
public static final int language44
```

language43

```
public static final int language43
```

language42

```
public static final int language42
```

language41

```
public static final int language41
```

languageBackgroundSoundCleanFeed

```
public static final int languageBackgroundSoundCleanFeed
```

charsetCompleteEBULatin

```
public static final int charsetCompleteEBULatin
```

charsetEBUCyrillicGreek

```
public static final int charsetEBUCyrillicGreek
```

charsetEBUArabic_HebrewETC

```
public static final int charsetEBUArabic_HebrewETC
```

charsetISOLatinAlphabetNo2

```
public static final int charsetISOLatinAlphabetNo2
```

serviceElementTypeUndefined

```
public static final int serviceElementTypeUndefined
```

serviceElementTypeEnsemble

```
public static final int serviceElementTypeEnsemble
```

serviceElementTypeService

```
public static final int serviceElementTypeService
```

serviceElementTypeComponent

```
public static final int serviceElementTypeComponent
```

locationInfoOnce

```
public static final int locationInfoOnce
```

locationInfoPeriodByTime

```
public static final int locationInfoPeriodByTime
```

locationInfoPeriodByDistance

```
public static final int locationInfoPeriodByDistance
```

locationInfoStop

```
public static final int locationInfoStop
```

locationInfoPosition

```
public static final int locationInfoPosition
```

locationInfoRegionId

```
public static final int locationInfoRegionId
```

operationControlSetVolume

```
public static final int operationControlSetVolume
```

operationControlGetVolume

```
public static final int operationControlGetVolume
```

operationControlSetServiceFollowing

```
public static final int operationControlSetServiceFollowing
```

operationControlGetServiceFollowing

```
public static final int operationControlGetServiceFollowing
```

operationControlGetServiceFollowingNotifications

```
public static final int operationControlGetServiceFollowingNotifications
```

operationControlSetDRCMMode

```
public static final int operationControlSetDRCMMode
```

operationControlGetDRCMMode

```
public static final int operationControlGetDRCMMode
```

operationControlGetDRCMModeNotifications

```
public static final int operationControlGetDRCMModeNotifications
```

serviceFollowingLeavingService

```
public static final int serviceFollowingLeavingService
```

serviceFollowingTryingAlternativeService

```
public static final int serviceFollowingTryingAlternativeService
```

serviceFollowingSelectingService

```
public static final int serviceFollowingSelectingService
```

streamTypeAudio

```
public static final int streamTypeAudio
```

streamTypePacket

```
public static final int streamTypePacket
```

streamTypeStream

```
public static final int streamTypeStream
```

streamTypeXPAD

```
public static final int streamTypeXPAD
```

streamTypeFIDC

```
public static final int streamTypeFIDC
```

subscriberInfoNoCA

```
public static final int subscriberInfoNoCA
```

subscriberInfoNoAlgorithm

```
public static final int subscriberInfoNoAlgorithm
```

subscriberInfoNoSubscription

```
public static final int subscriberInfoNoSubscription
```

subscriberInfoExpiredSubscription

```
public static final int subscriberInfoExpiredSubscription
```

conflictResolutionTurnProceed

```
public static final int conflictResolutionTurnProceed
```

conflictResolutionTurnProbe

```
public static final int conflictResolutionTurnProbe
```

conflictResolutionTurnStop

```
public static final int conflictResolutionTurnStop
```

conflictResolutionTurnPreempt

```
public static final int conflictResolutionTurnPreempt
```

conflictResolutionOperationNone

```
public static final int conflictResolutionOperationNone
```

conflictResolutionOperationTuneReq

```
public static final int conflictResolutionOperationTuneReq
```

conflictResolutionOperationSearchReq

```
public static final int conflictResolutionOperationSearchReq
```

conflictResolutionOperationScanReq

```
public static final int conflictResolutionOperationScanReq
```

conflictResolutionOperationSelectSIReq

```
public static final int conflictResolutionOperationSelectSIReq
```

conflictResolutionOperationGetEnsembleInfoReq

```
public static final int conflictResolutionOperationGetEnsembleInfoReq
```

conflictResolutionOperationGetServiceInfoReq

```
public static final int conflictResolutionOperationGetServiceInfoReq
```

conflictResolutionOperationGetComponentInfoReq

```
public static final int conflictResolutionOperationGetComponentInfoReq
```

conflictResolutionOperationSelectReceptionInfoReq

```
public static final int conflictResolutionOperationSelectReceptionInfoReq
```

conflictResolutionOperationSelectComponentReq

```
public static final int conflictResolutionOperationSelectComponentReq
```

conflictResolutionOperationSelectComponentStreamReq

```
public static final int conflictResolutionOperationSelectComponentStreamReq
```

conflictResolutionOperationSelectObjectReq

```
public static final int conflictResolutionOperationSelectObjectReq
```

conflictResolutionOperationGetLocationInfoReq

```
public static final int conflictResolutionOperationGetLocationInfoReq
```

conflictResolutionOperationOperationControlReq

```
public static final int conflictResolutionOperationOperationControlReq
```

conflictResolutionOperationSelectApplicationReq

```
public static final int conflictResolutionOperationSelectApplicationReq
```

conflictResolutionSuboperationNone

```
public static final int conflictResolutionSuboperationNone
```

conflictResolutionAnswerNo

```
public static final int conflictResolutionAnswerNo
```

conflictResolutionAnswerYes

```
public static final int conflictResolutionAnswerYes
```

Constructor Detail

DABConstants

```
public DABConstants()
```

Method Detail

result2String

```
public static java.lang.String result2String(int result)
```

The method returns a string for the given result code. It is an textual explanation of the result.

Class **DABException**

[dab](#)

java.lang.Object

|

+--java.lang.Throwable

|

+--java.lang.Exception

|

+--**dab.DABException**

All Implemented Interfaces:

java.io.Serializable

Direct Known Subclasses:

[DABConnectionException](#), [DABInvalidURLException](#), [DABNotAvailableException](#), [ResourceConflictException](#)

public Class **DABException**

extends java.lang.Exception

DABException is the superclass for exceptions inside the DAB package.

Version:

1.0

Constructor Summary	Page
DABException ()	286
DABException (java.lang.String msg)	286

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

DABException

public **DABException**()

DABException

```
public DABException(java.lang.String msg)
```

Interface DABListener[dab](#)**All Known Implementing Classes:**[DABAdapter](#)

```
public Interface DABListener
```

DABListener defines the interface for DAB listeners.

Note: to support J2ME MIDP profile DABListener does not implement java.util.EventListener

Version:

1.07

Method Summary		Page
void	componentNtf (dab.events.ComponentNtfEvent e) The componentNtf method is called if there are changes to the selection mode of a component.	349
void	conflictResolutionNtf (dab.events.ConflictResolutionNtfEvent e) The method is called for notifying the listener of resource conflicts.	350
void	drcModeNtf (dab.events.DRCModeNtfEvent e) The notification informs about DRC mode changes.	350
void	getComponentInfoCnf (dab.events.GetComponentInfoCnfEvent e) The GetComponentInfoCnf message finalizes the GetComponentInfo command and delivers information about a requested DAB Component to a DAB client.	348
void	getEnsembleInfoCnf (dab.events.GetEnsembleInfoCnfEvent e) The GetEnsembleInfoCnf method finalizes the GetEnsembleInfo command and delivers information about a requested DAB Ensemble to a DAB client.	348
void	getLocationInfoCnf (dab.events.GetLocationInfoCnfEvent e) getLocationInfoCnf confirms the getLocationInfo command.	350
void	getServiceInfoCnf (dab.events.GetServiceInfoCnfEvent e) A call to the getServiceInfoCnf method finalizes the GetServiceInfo command and delivers information about a requested DAB Service to a DAB client.	348
void	locationInfoNtf (dab.events.LocationInfoNtfEvent e) locationInfoNtf notifies about location information.	350
void	objectNtf (dab.events.ObjectNtfEvent e) The objectNtf method is called as a consequence of selecting objects from a data component by use of the SelectObject command.	349

void	<u>operationControlCnf</u> (dab.events.OperationControlCnfEvent e) The confirmation indicates the result of the operationControl command.	350
void	<u>receptionInfoNtf</u> (dab.events.ReceptionInfoNtfEvent e) The receptionInfoNtf method is called as a consequence of subscribing to state changes in synchronization, biterrorate and audio decoder muting.	349
void	<u>respondConflictResolutionCnf</u> (dab.events.RespondConflictResolutionCnfEvent e) The method is called for confirming a reaction to a resource conflict.	350
void	<u>scanCnf</u> (dab.events.ScanCnfEvent e) The ScanCnf message finalizes a Scan command.	347
void	<u>scanNtf</u> (dab.events.ScanNtfEvent e) The ScanNtf message is sent after a search for all available DAB Ensembles in a specified frequency range is started by the ScanReq message.	347
void	<u>searchCnf</u> (dab.events.SearchCnfEvent e) The searchCnf method finalizes a Search command and provides information about the command status, currently selected DAB frequency and current reception conditions.	347
void	<u>searchNtf</u> (dab.events.SearchNtfEvent e) A SearchNtf event is sent after a search for a DAB Ensemble was started searchReq.	347
void	<u>selectApplicationCnf</u> (dab.events.SelectApplicationCnfEvent e) The method is called as a consequence of selecting an application from a data component by use of the SelectApplication command.	349
void	<u>selectComponentCnf</u> (dab.events.SelectComponentCnfEvent e) The SelectComponentCnf confirmation finalizes the SelectComponent command.	349
void	<u>selectComponentStreamCnf</u> (dab.events.SelectComponentStreamCnfEvent e) The selectComponentStreamCnf method returns the requested stream (if the command was succesful) and informs about the result of the command.	349
void	<u>selectObjectCnf</u> (dab.events.SelectObjectCnfEvent e) The SelectObjectCnf method finalizes the SelectObject command.	349
void	<u>selectReceptionInfoCnf</u> (dab.events.SelectReceptionInfoCnfEvent e) The selectReceptionInfoCnf method finalizes the SelectReceptionInfo command.	348
void	<u>selectSICnf</u> (dab.events.SelectSICnfEvent e) The selectSICnf method finalizes a SelectSI command and indicates current settings.	348
void	<u>serviceFollowingNtf</u> (dab.events.ServiceFollowingNtfEvent e) The notification informs about service following actions.	350
void	<u>siNtf</u> (dab.events.SINtfEvent e) The siNtf notication is sent as a consequence of subscribing to Service Directory Information.	348
void	<u>systemFailureNtf</u> (dab.events.SystemFailureNtfEvent e) SystemFailureNtf notifies about severe problems with the hardware or the middleware (e.g.	351

void	tuneCnf (dab.events.TuneCnfEvent e)	347
The TuneCnf method finalizes a Tune command and is sent as a response to a TuneReq message.		

Method Detail

tuneCnf

```
public void tuneCnf(dab.events.TuneCnfEvent e)
```

The TuneCnf method finalizes a Tune command and is sent as a response to a TuneReq message. It provides information about the currently selected DAB frequency and reception conditions.

The Tune command is used to select a specified DAB frequency. The tuneReq request initiates the Tune command. tuneCnf finalizes the Tune command and provides information about the reception state. This includes the selected frequency, the detected transmission mode and the synchronization state of the receiver.

See Also:

[tuneReq](#), [searchReq](#), [searchCnf](#)

searchCnf

```
public void searchCnf(dab.events.SearchCnfEvent e)
```

The searchCnf method finalizes a Search command and provides information about the command status, currently selected DAB frequency and current reception conditions.

The Search command is used in order to search for a DAB Ensemble according to a specified search mode. Searching for a DAB Ensemble can take a large amount of time. The start of searching is indicated by a 'Started' searchNtf message. Other searchNtf messages inform a DAB client about search progress. It is finalized by delivery of the searchCnf message. It informs about the command status, the selected frequency and the synchronization state. No further searchNtf messages will be delivered after the delivery of the searchCnf message.

See Also:

[searchReq](#), [searchNtf](#)

searchNtf

```
public void searchNtf(dab.events.SearchNtfEvent e)
```

A SearchNtf event is sent after a search for a DAB Ensemble was started searchReq. It informs about the start of searching and about the progress of searching. A SearchCnf event finalizes a Search command. No more SearchNtf events are sent after a SearchCnf event was sent.

The SearchNtf event is sent after the searching for a DAB Ensemble has been started and while searching is in progress in order to provide information about the current status of searching. The 'Started' notification is sent in any case. Progress notifications are only sent if notifications have been requested with the related SearchReq message. No further notifications will be sent after a SearchCnf message is delivered.

See Also:

[searchReq](#), [searchCnf](#)

scanCnf

```
public void scanCnf(dab.events.ScanCnfEvent e)
```

The ScanCnf message finalizes a Scan command. It informs about the result of scanning and the current tune state.

The Scan command is used in order to perform a search for all available DAB Ensembles in a specified frequency range. Depending on the frequency range and the search mode this operation may require a substantial amount of time from only a second up to several minutes. The command is started by the ScanReq message and is finished with the ScanCnf message. In between ScanNtf messages are sent in order to inform about the current status of searching if notifications are requested.

The ScanCnf message indicates that the scan command is finished and informs about the current tune state. As a result of performing the scan command the service information database is filled with information. If a SI subscription is running several SINtf messages are delivered to the connected application.

See Also:

[scanReq](#), [scanNtf](#)

scanNtf

```
public void scanNtf(dab.events.ScanNtfEvent e)
```

The ScanNtf message is sent after a search for all available DAB Ensembles in a specified frequency range is started by the ScanReq message. The ScanNtf message provides information about the current status of searching for all available DAB Ensembles in a specified frequency range. It is delivered to the connected application after the search has been started by the ScanReq message and notifications have been requested. No further notifications will be sent after a ScanCnf message is delivered.

See Also:

[scanReq](#), [scanCnf](#)

selectSICnf

```
public void selectSICnf(dab.events.SelectSICnfEvent e)
```

The selectSICnf method finalizes a SelectSI command and indicates current settings. The SelectSI command starts, stops or changes subscription to Service Directory Information.

The SelectSI command allows a DAB client to subscribe for Service Directory Information. The subscription is requested by the selectSIREq method and is confirmed with the SelectSICnf method. The subscription level can be changed by another SelectSI command. This includes the termination of subscription. After a successful subscription a connected client receives siNtf calls when the Service Directory changes.

See Also:

[selectSIREq](#), [siNtf](#)

siNtf

```
public void siNtf(dab.events.SINtfEvent e)
```

The siNtf notification is sent as a consequence of subscribing to Service Directory Information.

A call to siNtf indicates that the Service Directory has changed. The type of change is signaled and a handle to the changed service element is provided. If AutoDelivery is activated the changed information object itself is delivered together with the notification. Otherwise it can be requested with getEnsembleInfo, getServiceInfo or getComponentInfo. siNtf message is called as a result of the subscription to Service Directory Information.

See Also:

[selectSIREq](#), [selectSICnf](#)

getEnsembleInfoCnf

```
public void getEnsembleInfoCnf(dab.events.GetEnsembleInfoCnfEvent e)
```

The GetEnsembleInfoCnf method finalizes the GetEnsembleInfo command and delivers information about a requested DAB Ensemble to a DAB client.

The GetEnsembleInfo command provides a DAB client with information about a specified DAB Ensemble, e.g. Label, No of Services, and so on. The command is initiated by a getEnsembleInfoReq request and is finished by a getEnsembleInfoCnf call.

See Also:

[getEnsembleInfoReq](#), [EnsembleInfo](#)

getServiceInfoCnf

```
public void getServiceInfoCnf(dab.events.GetServiceInfoCnfEvent e)
```

A call to the getServiceInfoCnf method finalizes the GetServiceInfo command and delivers information about a requested DAB Service to a DAB client.

The GetServiceInfo command provides a DAB client with information about a specified DAB Service, e.g. Label, No of Services, and so on. The command is initiated by a getServiceInfoReq message and is finished by a getServiceInfoCnf message.

See Also:

[getServiceInfoReq](#)

GetComponentInfoCnf

```
public void GetComponentInfoCnf(dab.events.GetComponentInfoCnfEvent e)
```

The GetComponentInfoCnf message finalizes the GetComponentInfo command and delivers information about a requested DAB Component to a DAB client.

The GetComponentInfo command provides a DAB client with information about a specified DAB Component, e.g. Label, Language and so on. The command is initiated by a GetComponentInfoReq request and is finished by a call to GetComponentInfoCnf message.

See Also:

[GetComponentInfoReq](#), [ComponentInfo](#)

selectReceptionInfoCnf

```
public void selectReceptionInfoCnf(dab.events.SelectReceptionInfoCnfEvent e)
```

The selectReceptionInfoCnf method finalizes the SelectReceptionInfo command. It informs about the command status and the current subscription level.

The selectReceptionInfo method allows a DAB client to subscribe for state change notifications concerning reception conditions in terms of synchronization, biterrorate and audio decoder muting. The subscription is requested by selectReceptionInfoReq and is confirmed with selectReceptionInfoCnf. The subscription level can be changed by another SelectReceptionInfo command. This includes stopping of subscription. After a successful subscription the calling DAB client receives receptionInfoNtf calls when state changes occur.

See Also:

[selectReceptionInfoReq](#), [receptionInfoNtf](#)

receptionInfoNtf

```
public void receptionInfoNtf(dab.events.ReceptionInfoNtfEvent e)
```

The receptionInfoNtf method is called as a consequence of subscribing to state changes in synchronization, biterrorate and audio decoder muting.

receptionInfoNtf indicates that the synchronization state, biterrorate or mute state has changed (cf. ReceptionInfoNtfEvent). The ReceptionInfoNtf message is provided to a connected client as a result of subscription to state change notifications concerning reception conditions (selectReceptionInfoReq and selectReceptionInfoCnf messages).

See Also:

[selectReceptionInfoReq](#), [selectReceptionInfoCnf](#)

selectComponentCnf

```
public void selectComponentCnf(dab.events.SelectComponentCnfEvent e)
```

The SelectComponentCnf confirmation finalizes the SelectComponent command. It informs about the command status and the selection status of the specified component.

The SelectComponent command allows to start or stop applications delivered in DAB components. In general more than one component of the same DAB Ensemble can be selected simultaneously. It is possible to select one audio component, all programme-associated data components of the selected audio component and more than one independent data component at the same time. The selection of a component is requested by the selectComponentReq message and is confirmed by a selectComponentCnf call. It is possible that a component is removed from a DAB Ensemble which means it is no longer broadcast and therefore no longer available. This is indicated by a siNtf call and means that the selection is removed automatically.

See Also:

[selectComponentReq](#), [siNtf](#)

selectComponentStreamCnf

```
public void selectComponentStreamCnf(dab.events.SelectComponentStreamCnfEvent e)
```

The selectComponentStreamCnf method returns the requested stream (if the command was successful) and informs about the result of the command.

See Also:

[selectComponentStreamReq](#)

componentNtf

```
public void componentNtf(dab.events.ComponentNtfEvent e)
```

The componentNtf method is called if there are changes to the selection mode of a component. This typically happens, when the selection of a component is stopped.

Note, that this notification will be produced due to internal reasons (e.g. after tuning to another ensemble) and not in response to a selectComponentReq request (that is handled by selectComponentCnf).

selectObjectCnf

```
public void selectObjectCnf(dab.events.SelectObjectCnfEvent e)
```

The SelectObjectCnf method finalizes the SelectObject command. The SelectObject command selects an object from a selected DAB Component. This includes requesting an object from a data component, delivery after reception and notification of updates as long as the object is selected.

The SelectObject command selects an object from a selected component. Selection means it is requested for delivery and if wanted also updates of the object are delivered. Additionally it is possible to give some hints for caching. More than one object and also from more than one component can be selected simultaneously. The selection of an object is requested by selectObjectReq and is confirmed by calling selectObjectCnf. The object is delivered using objectNtf. This includes first-time delivery and all updates. Beyond starting or stopping a selection it is possible to remove all other selections belonging to the same component by setting parameter replaceSelections to true. It is possible that a component is removed from a DAB Ensemble. This is indicated by a call to siNtf. In this case also the selected objects of the service are no longer selected. It is possible that an object is removed from current on-air service. This is indicated by calling objectNtf. In this case the selections for this object are automatically disabled. Currently object selection makes only sense with applications of type BroadcastWebSite. Objects of applications like Slideshows or Dynamic Label are delivered automatically using objectNtf.

See Also:

[selectObjectReq](#), [objectNtf](#)

selectApplicationCnf

```
public void selectApplicationCnf(dab.events.SelectApplicationCnfEvent e)
```

The method is called as a consequence of selecting an application from a data component by use of the SelectApplication command. It delivers a proxy to control the application.

See Also:

[selectApplicationReq](#)

objectNtf

```
public void objectNtf(dab.events.ObjectNtfEvent e)
```

The objectNtf method is called as a consequence of selecting objects from a data component by use of the SelectObject command. It delivers a selected object partially or complete to a DAB client.

objectNtf is used to deliver a selected object to the connected DAB client. Depending on the request mode the object is delivered only once or more than once in case of updates. If the object can not be delivered in-time as indicated by a call to selectObjectCnf, then objectNtf informs about the delay. If transmission of an selected object is stopped, objectNtf informs about the termination of the object transmission and the object selection. It is possible that a DAB Component is removed from a DAB Ensemble. This is indicated by a call to siNtf. In this case also the selected objects of the component are no longer selected. No termination messages are sent for terminated object selections resulting from termination of a component.

See Also:

[selectObjectReq](#), [selectObjectCnf](#), [DABObject](#)

getLocationInfoCnf

```
public void getLocationInfoCnf(dab.events.GetLocationInfoCnfEvent e)
```

getLocationInfoCnf confirms the getLocationInfo command. This means the delivery of location information will start from now on.

See Also:

[getLocationInfoReq](#), [locationInfoNtf](#)

locationInfoNtf

```
public void locationInfoNtf(dab.events.LocationInfoNtfEvent e)
```

locationInfoNtf notifies about location information.

See Also:

[getLocationInfoReq](#), [getLocationInfoCnf](#)

conflictResolutionNtf

```
public void conflictResolutionNtf(dab.events.ConflictResolutionNtfEvent e)
```

The method is called for notifying the listener of resource conflicts. The listener can react to this event using the request respondConflictResolutionReq.

See Also:

[respondConflictResolutionReq](#), [respondConflictResolutionCnf](#)

respondConflictResolutionCnf

```
public void respondConflictResolutionCnf(dab.events.RespondConflictResolutionCnfEvent e)
```

The method is called for confirming a reaction to a resource conflict.

See Also:

[respondConflictResolutionReq](#), [conflictResolutionNtf](#)

operationControlCnf

```
public void operationControlCnf(dab.events.OperationControlCnfEvent e)
```

The confirmation indicates the result of the operationControl command.

See Also:

[operationControlReq](#)

serviceFollowingNtf

```
public void serviceFollowingNtf(dab.events.ServiceFollowingNtfEvent e)
```

The notification informs about service following actions.

See Also:

[operationControlReq](#)

drcModeNtf

```
public void drcModeNtf(dab.events.DRCModeNtfEvent e)
```

The notification informs about DRC mode changes.

See Also:

[operationControlReq](#)

systemFailureNtf

```
public void systemFailureNtf(dab.events.SystemFailureNtfEvent e)
```

SystemFailureNtf notifies about severe problems with the hardware or the middleware (e.g. breakdown of the communication to the DAB receiver). This should not be confused with the indication of errors for a particular command, which relates only to the command itself.

Typically, after the notification is sent, the package can no longer be used or needs to be reinitialised.

Class **DABNotAvailableException**

[dab](#)

```

java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--dab.DABException
            |
            +--dab.DABNotAvailableException
  
```

All Implemented Interfaces:

java.io.Serializable

public Class **DABNotAvailableException**

extends [DABException](#)

The *DABNotAvailableException* is thrown when particular data is currently not available or even not at all available. This usually happens with respect to so-called optional attributes.

Version:

1.0

Constructor Summary	Page
DABNotAvailableException()	277

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

DABNotAvailableException

```
public DABNotAvailableException()
```

Class *DABReceiverAddress*

[dab](#)

java.lang.Object

|

+--**dab.DABReceiverAddress**

public Class **DABReceiverAddress**

extends java.lang.Object

DABReceiverAddress is used to specify the location of DAB receivers.

Version:

1.0

Constructor Summary		Page
DABReceiverAddress (java.lang.String address)	Generates a DABReceiverAddress object from a textual representation of the address.	276

Method Summary		Page
java.lang.String getAddress ()	Generates a textual representation of the object, that can be used to construct a DABReceiverAdress object	276
java.lang.String toString ()	Generates a textual representation of the object.	389

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

DABReceiverAddress

public **DABReceiverAddress**(java.lang.String address)

Generates a DABReceiverAddress object from a textual representation of the address. Note, that the actual format of address depends on the implementation

See Also:

[getAddress](#)

Method Detail

toString

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

Generates a textual representation of the object.

Overrides:

toString in class java.lang.Object

getAddress

public java.lang.String **getAddress**()

Generates a textual representation of the object, that can be used to construct a DABReceiverAdress object

Interface DABSource

[dab](#)

All Known Implementing Classes:

[DABClient](#)

public Interface **DABSource**

DABSource defines the interface of a DAB resource (usually a DAB receiver). The interface is asynchronous. When an application issues a requests, it gets back the results as confirmation and notification events. Look at the particular methods for more details.

Version:

1.07

See Also:

[DABListener](#)

Method Summary		Page
void	getComponentInfoReq (dab.si.ComponentId componentId) The getComponentInfoReq request initiates a GetComponentInfo command.	338
void	getEnsembleInfoReq (dab.si.EnsembleId id) The getEnsembleInfoReq method initiates a GetEnsembleInfo command.	338
void	getLocationInfoReq (int type, int mode, int desiredDelta, int desiredAccuracy) The getLocationInfoReq initiates the GetLocationInfoCommand.	340
void	getServiceInfoReq (dab.si.ServiceId id) The getServiceInfoReq requests initiates a GetServiceInfo command.	338
void	operationControlReq (int attribute, java.lang.Object value) The OperationControl command enables the DAB client to change or read receiver parameters.	340
void	respondConflictResolutionReq (int transaction, int turn, int operation, int suboperation, int answer) The respondConflictResolutionReq is used to respond to a resource conflict notification.	340

void	scanReq (int searchMode, int tables, int startFrequency, int stopFrequency, int transmissionModes, int notifications) The ScanReq request initiates a Scan command.	338
void	searchReq (int searchMode, int tables, int startFrequency, int stopFrequency, int transmissionModes, int notifications) The searchReq request initiates a Search command.	337
void	selectApplicationReq (dab.si.ComponentId serviceId, dab.data.ObjectId objectId) The selectApplicationReq selects applications The SelectApplication command enables a client to load and control an application.	339
void	selectComponentReq (dab.si.ComponentId id, int selectionMode) The selectComponentReq request initiates the SelectComponent command.	339
void	selectComponentStreamReq (dab.si.ComponentId componentId) The following request provides access to the DAB transport streams.	339
void	selectObjectReq (dab.si.ComponentId id, dab.data.ObjectId objectId, int requestMode, boolean replaceSelections, int deliveryMode, int cacheHint) The selectObjectReq request initiates the SelectObject command.	339
void	selectReceptionInfoReq (boolean synchronizationNotification, boolean bitErrorRateNotifications, boolean muteStateNotifications, boolean requestOnce) The selectReceptionInfoReq request initiates the SelectReceptionInfo command.	339
void	selectSIRReq (boolean ensembleInfo, boolean serviceInfo, boolean componentInfo, boolean autoDelivery) The selectSIRReq method initiates a SelectSI command.	338
void	tuneReq (int tuneFrequency, int transmissionMode) The tuneReq request initiates the Tune command.	337

Method Detail

tuneReq

```
public void tuneReq(int tuneFrequency,
                    int transmissionMode)
    throws DABException,
           java.lang.SecurityException
```

The tuneReq request initiates the Tune command. The Tune command sets directly a specified DAB frequency. A DAB receiver must be tuned to a DAB frequency and synchronized in order to get access to DAB services. A tuned DAB receiver tries automatically to synchronize on a DAB Ensemble.

The Tune command is used to select a specified DAB frequency. The tuneReq request initiates the Tune command. Depending on the specification for the Transmissionmode it is tested if a DAB Ensemble can be detected. If the connected DAB receiver supports automatic detection the default setting for transmissionMode (=DABConstants.transmissionModeAutomatic) can be used. Otherwise it has to be specified which transmission modes should be tested. The result of the command is delivered by the tuneCnf confirmation. All currently existing selections

of audio and data services or selections of data objects are automatically stopped before tuning is performed by the DAB receiver.

Parameters:

tuneFrequency - This parameter specifies the frequency the DAB receiver will be tuned to in Hertz.
transmissionModes - This parameter specifies the transmission modes a DAB receiver tests for DAB Ensembles. The default value is `DABConstants.transmissionModeAutomatic` which means that the receiver is automatically detecting the transmission mode. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.transmissionModeAutomatic`: The transmission mode is automatically detected. All other flags are ignored in this case.
- `DABConstants.transmissionMode1`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 1.
- `DABConstants.transmissionMode2`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 2.
- `DABConstants.transmissionMode3`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 3.
- `DABConstants.transmissionMode4`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 4.

Throws:

[DABException](#) - in case of invalid DAB parameter
[SecurityException](#) - unauthorized usage of `tuneReq`

See Also:

[tuneCnf](#), [searchReq](#), [searchCnf](#), [searchNtf](#)

searchReq

```
public void searchReq(int searchMode,
                    int tables,
                    int startFrequency,
                    int stopFrequency,
                    int transmissionModes,
                    int notifications)
    throws DABException,
           java.lang.SecurityException
```

The `searchReq` request initiates a Search command. The Search command searches for a DAB Ensemble according to a specified search mode. After a successful execution of the Search command a DAB Ensemble has been found, the state Tuned is entered and the DAB receiver tries to synchronize automatically to the found DAB Ensemble.

The Search command is used to search for a DAB Ensemble. The `searchReq` request initiates the search and specifies the frequencies and transmission modes to test. Additionally the notifications can be specified which the DAB client gets while the command is executed. Searching for an ensemble may require a substantial amount of time from only a second up to several minutes. This depends also on the search mode specified. If the reception conditions are bad it is possible that no DAB Ensemble at all is detected. In order to stop searching for a DAB Ensemble the Tune command can be used which tunes the DAB Receiver to a certain frequency independent from the reception conditions. The start of searching is indicated by a `SearchNtf` event with a status code 'Started'. In this case the state machine of Tune State enters the state Searching (see Figure 4). In case that the previous state was Tuned all currently existing selections of services or objects are stopped automatically. While searching is performed several notifications delivering information about the current status are sent to the client. The command ends with a `SearchCnf` event.

Parameters:

searchMode - This parameter specifies the way the DAB receiver is searching for a DAB Ensemble. The default value is `searchSearchAutomatic` which means it is searching according to a default method. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.SearchModeAutomatic`: default method
- `DABConstants.SearchMode16kHzSteps`: The frequency range is searched in 16 kHz steps. This is a very intensive search which means that command execution can take a large amount of time.
- `DABConstants.SearchModeUp`: The search direction is from low to high frequencies.
- `DABConstants.SearchModeDown`: The search direction is from high to low frequencies.
- `DABConstants.SearchModeUseTables`: The search is based on the specified frequency tables.
- `DABConstants.SearchModeUseFrequencyRange`: The search is based on the specified frequency range.
- `DABConstants.SearchModeContinuous`: The search is looping over the specified frequency range until a DAB Ensemble has been found. The default is to stop after the specified frequency range has been checked once.

`tables` - This parameter specifies frequency tables the receiver uses in order to search for DAB Ensembles. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.searchCEPTFrequencyTableBandIII`: The frequencies according to the CEPT frequency table for Band III are tested for DAB Ensembles.
- `DABConstants.SearchCEPTFrequencyTableLBand`: The frequencies according to the CEPT L-Band table are tested for DAB Ensembles.
- `DABConstants.SearchCanadaFrequencyTableLBand`: The frequencies according to the Canadian L-Band table are tested for DAB Ensembles.

`transmissionModes` - This parameter specifies the transmission modes a DAB receiver tests for DAB Ensembles. The default value is `DABConstants.transmissionModeAutomatic` which means that the receiver is automatically detecting the transmission mode. The parameter is a flag field supporting the following flags which can be specified together:

- `transmissionModeAutomatic`: The transmission mode is automatically detected. All other flags are ignored.
- `transmissionMode1`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 1.
- `transmissionMode2`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 2.
- `transmissionMode3`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 3.
- `transmissionMode4`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 4.

`notifications` - This parameter specifies the type of notifications the client wants to get while the Seek command is performed. The parameter is a flag field supporting the following flags which can be specified together:

- `notificationsOff`: No intermediate notifications are sent. Only a `SearchNtf` notification which informs about the start of searching is sent.
- `notifications16kHzSteps`: With each 16 kHz step a notification is sent. This is only possible if 16kHz step searching is specified as search mode.
- `notificationsTableEntry`: With each table entry frequency a notification is sent. This is the default value.

See Also:

[searchCnf](#), [searchNtf](#), [tuneReq](#), [tuneCnf](#)

scanReq

```
public void scanReq(int searchMode,
                    int tables,
                    int startFrequency,
                    int stopFrequency,
                    int transmissionModes,
                    int notifications)
    throws DABException,
           java.lang.SecurityException
```

The `ScanReq` request initiates a Scan command. The Scan command is used in order to perform a search for all available DAB Ensembles in a specified frequency range. Depending on the frequency range and the search mode this operation may require a substantial amount of time from only a second up to several minutes. The command is started by the `ScanReq` request and is finished with the `ScanCnf` confirmation. In between `ScanNtf` notification are sent in order to inform about the current status of scanning if notifications are requested.

In case of searching from lower to higher frequencies (`searchMode=DABConstants.searchModeUp`) the value of `startFrequency` is not allowed to be larger than the value of `stopFrequency`. In case of searching from higher to lower frequencies (`searchMode=DABConstants.searchModeDown`) the value of `startFrequency` is not allowed to be smaller than the value of `stopFrequency`.

Parameters:

`searchMode` - This parameter specifies the way the DAB Receiver is searching for a DAB Ensemble. The default value is `DABConstants.searchModeAutomatic` which means it is searching according to a default method. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.searchModeAutomatic`: default method
- `DABConstants.searchMode16kHzSteps`: The frequency range is searched in 16 kHz steps.
- `DABConstants.searchModeUp`: The search direction is from low to high frequencies.
- `DABConstants.searchModeDown`: The search direction is from high to low frequencies.
- `DABConstants.searchModeUseTables`: The search is based on the specified frequency tables.
- `DABConstants.searchModeUseFrequencyRange`: The search is based on the specified frequency range.
- `DABConstants.searchModeContinuous`: The search is looping over the specified frequency range until a DAB Ensemble has been found. The default is to stop after the specified frequency range has been checked once.

`tables` - This parameter specifies frequency tables the receiver uses in order to search for DAB Ensembles. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.searchCEPTFrequencyTableBandIII`: The frequencies according to the CEPT frequency table for Band III are tested for DAB Ensembles.
- `DABConstants.searchCEPTFrequencyTableLBand`: The frequencies according to the CEPT L-Band table are tested for DAB Ensembles.
- `DABConstants.searchCanadaFrequencyTableLBand`: The frequencies according to the Canadian L-Band table are tested for DAB Ensembles.

`startFrequency` - This parameter specifies the start frequency at which the DAB Receiver starts its search for DAB Ensembles.

`stopFrequency` - This parameter specifies the stop frequency at which the DAB Receiver stops its search for DAB Ensembles.

`transmissionModes` - This parameter specifies the transmission modes a DAB Receiver should look for DAB Ensembles. The default value is `DABConstants.transmissionModeAutomatic` which means that the receiver is automatically detecting the transmission mode. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.transmissionModeAutomatic`: The transmission mode is automatically detected.
- `DABConstants.transmissionMode1`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 1.
- `DABConstants.transmissionMode2`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 2.
- `DABConstants.transmissionMode3`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 3.
- `DABConstants.transmissionMode4`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 4.

`notifications` - This parameter specifies the type of notifications wanted by the application while the Seek command is performed. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.notificationsOff`: No notifications are sent.
- `DABConstants.notifications16kHzSteps`: With each 16 kHz step a notification is sent.
- `DABConstants.notificationsTableEntry`: With each table entry frequency a notification is sent. This is the default value.

See Also:

[scanCnf](#), [scanNtf](#)

selectSIReq

```
public void selectSIReq(boolean ensembleInfo,  
                        boolean serviceInfo,  
                        boolean componentInfo,  
                        boolean autoDelivery)  
    throws DABException,  
           java.lang.SecurityException
```

The `selectSIReq` method initiates a SelectSI command. The SelectSI command starts, stops or changes subscription to Service Directory Information.

The SelectSI command allows a DAB client to subscribe for Service Directory Information. The Service Directory contains all available ensembles, services, components and related information. The subscription is requested by the selectSIReq request and is confirmed with the selectSICnf confirmation. The subscription level can be changed by another SelectSI command. This includes the termination of subscription.

After a successful subscription a connected client receives SINtf notifications. Just after the subscription has been activated the complete content of the Service Directory is mapped on SINtf notifications. This means for each stored instance of a service element (ensemble, service and component) is a SINtf notification sent which indicates that this service element is available (Added). As time goes on SINtf notifications are sent which indicate that a new service element is available (Added), that an existing is no longer available (Removed) or that its attributes have changed (Changed).

By use of autoDelivery it can be specified if SINtf sends only a notification or a notification together with the related information object. If subscription is terminated by setting ensembleInfo, serviceInfo and componentInfo to false, then autoDelivery has no meaning.

By selecting a certain subscription level the client is informed about all currently known service elements by sending related SINtf notifications. This means if a client subscribes for service-specific notifications and seven services exist at this time, then seven SINtf(DABConstants.serviceAdded) notifications are generated. The client is not informed about known ensembles or components. As time goes on the client is informed when new services are added, known services are removed or changed. If a currently selected subscription level is increased meaning that more notification types are subscribed then the client is informed about all currently known service elements that are related to the new subscribed notification type. This means if a subscription is changed from service-specific to service-specific and component-specific change notifications, then for each currently known Component a SINtf(DABConstants.componentAdded) notification is generated.

As time goes on the client is informed when new services or components are added, known services or components are removed or changed. If a currently selected subscription level is decreased meaning that less notification types are subscribed then the client is informed only about notifications related to the remaining subscribed notification types. This means if a subscription is changed from service-specific and component-specific to service-specific notifications, then the client is informed when new services are added, known services are removed or changed. But the client is no longer notified about changes related to components.

Parameters:

ensembleInfo - This parameter specifies if ensemble-specific notifications will be sent to the DAB client. The following values are supported:

- true: The DAB client is notified about DABConstants.ensembleAdded, DABConstants.ensembleChanged and DABConstants.ensembleRemoved events. This is the default setting.
- false: The DAB client is not notified about DABConstants.ensembleAdded, DABConstants.ensembleChanged and DABConstants.ensembleRemoved events.

serviceInfo - this parameter specifies if service-specific notifications will be sent to the DAB client. The following values are supported:

- true: The DAB client is notified about DABConstants.serviceAdded, DABConstants.serviceChanged and DABConstants.serviceRemoved events. This is the default setting.
- false: The DAB client is not notified about DABConstants.serviceAdded, DABConstants.serviceChanged and DABConstants.serviceRemoved events.

componentInfo - This parameter specifies if component-specific notifications will be sent to the DAB client. The following values are supported:

- true: The DAB client is notified about DABConstants.componentAdded, DABConstants.componentChanged and DABConstants.componentRemoved events. This is

the default setting.

- false: The DAB client is not notified about `DABConstants.componentAdded`, `DABConstants.componentChanged` and `DABConstants.componentRemoved` events.

`autoDelivery` - This parameter specifies if the information related to the notification is sent together with the notification (SINtf) or not. The following values are supported:

- true: The SINtf notification delivers the notification together with the information object. The information object is sent together with the notification if the notification type is -Added or -Changed. In case of -Removed no information object is sent because it is no longer existing. This is the default setting.
- false: The SINtf notification delivers only the notification. The information object (`EnsembleInfo`, `ServiceInfo` or `ComponentInfo`) itself can be obtained by use of `getEnsembleInfoReq`, `getServiceInfoReq` or `getComponentInfoReq`.

See Also:

[selectSICnf](#), [siNtf](#)

getEnsembleInfoReq

```
public void getEnsembleInfoReq(dab.si.EnsembleId id)
    throws DABException,
           java.lang.SecurityException
```

The `getEnsembleInfoReq` method initiates a `GetEnsembleInfo` command. The `GetEnsembleInfo` command requests information about the specified DAB Ensemble.

The `GetEnsembleInfo` command provides a DAB client with information about a specified DAB Ensemble, e.g. Label, No of Services, and so on. The command is initiated by a `getEnsembleInfoReq` request and is finished by a `getEnsembleInfoCnf` confirmation.

Parameters:

`id` - This parameter is a handle identifying the DAB Ensemble.

See Also:

[getEnsembleInfoCnf](#)

getServiceInfoReq

```
public void getServiceInfoReq(dab.si.ServiceId id)
    throws DABException,
           java.lang.SecurityException
```

The `getServiceInfoReq` requests initiates a `GetServiceInfo` command. The `GetServiceInfo` command requests information about a specified DAB Service.

The `GetServiceInfo` command provides a DAB client with information about a specified DAB Service, e.g. Label, No of Components, and so on. The command is initiated by a `getServiceInfoReq` request and is finished by a `getServiceInfoCnf` confirmation.

Parameters:

`id` - This parameter is a handle identifying the DAB Service.

See Also:

[getServiceInfoCnf](#)

getComponentInfoReq

```
public void getComponentInfoReq(dab.si.ComponentId componentId)
    throws DABException,
           java.lang.SecurityException
```

The `getComponentInfoReq` request initiates a `GetComponentInfo` command. The `GetComponentInfo` command requests information about a specified DAB Component.

The `GetComponentInfo` command provides a DAB client with information about a specified DAB Component, e.g. Label, Language and so on. The command is initiated calling `getComponentInfoReq` and is finished by a call to `getComponentInfoCnf`.

Parameters:

`serviceId` - This parameter is a handle identifying the DAB Component.

See Also:

[getComponentInfoCnf](#)

selectReceptionInfoReq

```
public void selectReceptionInfoReq(boolean synchronizationNotification,
    boolean bitErrorRateNotifcations,
    boolean muteStateNotifcations,
    boolean requestOnce)
    throws DABException,
           java.lang.SecurityException
```

The `selectReceptionInfoReq` request initiates the `SelectReceptionInfo` command. The `SelectReceptionInfo` command starts, stops or changes subscription to state change notifications concerning reception conditions. It is possible to monitor synchronization, biterrorrate and audio decoder muting.

The `SelectReceptionInfo` command allows a DAB client to subscribe for state change notifications concerning reception conditions in terms of synchronization, biterrorrate and audio decoder muting. The subscription is requested by the `selectReceptionInfoReq` request and is confirmed with the `selectReceptionInfoCnf` confirmation. The subscription level can be changed by another `SelectReceptionInfo` command. This includes stopping of subscription. After a successful subscription the calling DAB client receives `ReceptionInfoNtf` notifications when state changes occur.

Parameters:

`synchronizationNotification` - This parameter specifies if the calling client is notified about state changes concerning DAB signal synchronization. If the parameter is set to true (default) notifications are provided, if it is set to false no notifications are provided.

`bitErrorRateNotifcations` - This parameter specifies if the calling client is notified about state changes concerning the biterrorrate. If the parameter is set to true (default) notifications are provided, if it is set to false no notifications are provided.

`muteStateNotifcations` - This parameter specifies if the calling client is notified about state changes concerning the mute state of the audio decoder. If the parameter is set to true (default) notifications are provided, if it is set to false no notifications are provided.

`requestOnce` - This parameter specifies if the reception condition information is wanted only once. In this case the reception condition is once detected and the DAB client informed by one and only one `receptionInfoNtf` call.

See Also:

[selectReceptionInfoCnf](#), [receptionInfoNtf](#)

selectComponentReq

```
public void selectComponentReq(dab.si.ComponentId id,
    int selectionMode)
    throws DABException,
           java.lang.SecurityException
```

The `selectComponentReq` request initiates the `SelectComponent` command. The `SelectComponent` command starts or stops an application delivered in a DAB Component.

The `SelectComponent` command allows to start or stop applications delivered in DAB components. In general more than one component of the same DAB Ensemble can be selected simultaneously. It is possible to select one audio component, all programme-associated data components of the selected audio component and more than one independent data component at the same time. The selection of a component is requested by the `selectComponentReq` request and is confirmed by the `selectComponentCnf` confirmation. It is possible that a component is removed from a DAB Ensemble which means it is no longer broadcast and therefore no longer available. This is indicated by a `SINtf` call and means that the selection is removed automatically. If the selection of a component is removed also all existing object selections belonging to the component are removed.

If the user application is a slide show or a dynamic label, its objects are delivered automatically (using `objectNtf` notifications) after the `SelectComponent` confirmation was sent.

If the selected component is an audio service, its PAD data services become available as "virtual components". This means service information is generated for all PAD services and they can be selected. If the selection of the audio service is stopped, also all PAD services are stopped and they are not available anymore.

If the component is not in the current ensemble, it depends on the implementation whether it is selected nevertheless.

Parameters:

`id` - This parameter is a pointer to the identifier of the DAB Component which is to be selected. If all component selections should be removed (set `selectionMode` to `DABConstants.selectionModeRemoveAll`) this parameter is ignored and should be set to null.

`selectionMode` - This parameter specifies the selection mode for the component. The following flags are supported:

- `DABConstants.selectionModeReplace`: All currently selected components of the same type are stopped and the specified component is to be started. The same type means an audio component replaces any other selected audio component, a data component replaces all other selected independent data components and a programme-associated data component replaces all other selected programme-associated data components.
- `DABConstants.selectionModeAdd`: The application delivered by the specified component is to be started. Other selected components are not affected.
- `DABConstants.selectionModeRemove`: The selection of the specified component is stopped.
- `DABConstants.selectionModeRemoveAll`: All existing component selections are removed. Set `serviceld` to null in this case.

See Also:

[selectComponentCnf](#), [siNtf](#), [serviceObjectReq](#)

`selectComponentStreamReq`

```
public void selectComponentStreamReq(dab.si.ComponentId componentId)
    throws DABException,
           java.lang.SecurityException
```

The following request provides access to the DAB transport streams. The requested stream is delivered back in the confirmation, which ends the command.

Parameters:

`componentId` - the service identifier of the component which carries the stream

See Also:

[selectComponentStreamCnf](#)

selectApplicationReq

```
public void selectApplicationReq(dab.si.ComponentId serviceId,  
                                dab.data.ObjectId objectId)  
    throws DABException,  
           java.lang.SecurityException
```

The selectApplicationReq selects applications

The SelectApplication command enables a client to load and control an application. The request is confirmed with the selectApplicationCnf confirmation.

Parameters:

`serviceId` - the component in which the application is located
`objectId` - the id of the start object

See Also:

[selectApplicationCnf](#)

selectObjectReq

```
public void selectObjectReq(dab.si.ComponentId id,  
                             dab.data.ObjectId objectId,  
                             int requestMode,  
                             boolean replaceSelections,  
                             int deliveryMode,  
                             int cacheHint)  
    throws DABException,  
           java.lang.SecurityException
```

The selectObjectReq request initiates the SelectObject command. The SelectObject command selects an object from a selected DAB Component. This includes requesting an object from a data component, delivery after reception and notification of updates as long as the object is selected.

The SelectObject command selects an object from a selected component. Selection means it is requested for delivery and if wanted also updates of the object are delivered. Additionally it is possible to give some hints for caching. More than one object and also from more than one component can be selected simultaneously. The selection of an object is requested by the selectObjectReq request and is confirmed by the selectObjectCnf confirmation. The object is delivered using the objectNtf method. This includes first-time delivery and all updates. Beyond starting or stopping a selection it is possible to remove all other selections belonging to the same component by setting parameter replaceSelections to true. It is possible that a component is removed from a DAB Ensemble. This is indicated by a serviceInfoNtf call. In this case also the selected objects of the service are no longer selected.

It is possible that an object is removed from current on-air service. This is indicated by an objectNtf call. In this case the selections for this object are automatically disabled.

Currently object selection makes only sense with applications of type BroadcastWebSite. Objects of applications like Slideshows or Dynamic Label are delivered automatically by objectNtf calls.

Parameters:

`id` - This parameter identifies the selected component the object is belonging to.
`objectId` - This parameter identifies the object which is to be selected.
`selectionMode` - This parameter specifies the selection mode of the object. The following values are supported:

- `DABConstants.requestModeOff`: This is used in order to stop the selection of objects which are requested with the request mode `DABConstants.requestModeUpdate`. It is not needed for objects which are requested with the `DABConstants.requestModeOnce` flag except for the case that a `SelectObjectReq` is pending and the delivery is no longer wanted.
- `DABConstants.requestModeOnce`: The object is requested for one-time delivery. After the first reception from the broadcast channel the object is delivered to the connected DAB client. The client is not notified about new versions.

- `DABConstants.requestModeUpdate`: The object is requested for update delivery. After the first reception from the broadcast channel the object is delivered to the connected client. Additionally each new version of the object is delivered.

`replaceSelections` - This parameter specifies if all current object selections belonging to the component identified by `serviceId` are replaced with this selection. If this parameter is set to true, then all selections are removed. If this parameter is set to false, then existing selections remain unchanged.

`deliveryMode` - This parameter specifies the delivery mode of the object. The following values are supported:

- `DABConstants.deliveryModeComplete`: Only the complete object is delivered to the DAB client.
- `DABConstants.deliveryModePartial`: The object may be delivered in parts.

`cacheHint` - This parameter specifies a hint for caching of the selected object.

See Also:

[selectObjectCnf](#), [objectNtf](#)

getLocationInfoReq

```
public void getLocationInfoReq(int type,
                               int mode,
                               int desiredDelta,
                               int desiredAccuracy)
    throws DABException,
           java.lang.SecurityException
```

The `getLocationInfoReq` initiates the `GetLocationInfoCommand`.

Parameters:

`type` - This parameter indicates the type of location information, that is requested. Supported flags are `DABConstants.LocationInfoPosition` and `DABConstants.LocationInfoRegionId`.

`mode` - The information is delivered according to the values of this parameter:

- `DABConstants.LocationInfoOnce`: the information is delivered only one time. The parameter `desiredDelta` is not considered.
- `DABConstants.LocationInfoPeriodByTime`: The information is delivered in intervals given by the value of `desiredDelta` (in milliseconds)
- `DABConstants.LocationInfoPeriodByDistance`: The information is delivered after the distance has passed given by the value of `desiredDelta` (in meters)
- `DABConstants.LocationInfoStop`: The delivery of information is stopped. The parameter `desiredDelta` is not considered.

`desiredDelta` - cf. description of the mode parameter

`desiredAccuracy` - This parameter indicates the desired accuracy in meters. The value is only considered if `type&DABConstants.LocationInfoPosition!=0`

This command is optional and may only partially be supported (e.g. only `mode=DABConstants.LocationInfoOnce` and `mode=DABConstants.LocationInfoStop`) or may not be supported at all.

See Also:

[getLocationInfoCnf](#), [locationInfoNtf](#)

respondConflictResolutionReq

```
public void respondConflictResolutionReq(int transaction,
                                         int turn,
                                         int operation,
                                         int suboperation,
                                         int answer)
```

The `respondConflictResolutionReq` is used to respond to a resource conflict notification.

Parameters:

`transaction` - the identifier of the transaction of the resource conflict
`turn` - the code of the turn (cf. `DABConstants.conflictResolutionTurn*`)
`operation` - the code of the operation (cf. `DABConstants.conflictResolutionOperation*`)
`suboperation` - the code of the suboperation (cf. `DABConstants.conflictResolutionSuboperation*`)
`answer` - the actual answer (cf. `DABConstants.conflictResolutionAnswer*`)

See Also:

[conflictResolutionNtf](#), [respondConflictResolutionCnf](#)

operationControlReq

```
public void operationControlReq(int attribute,
                                java.lang.Object value)
    throws DABException,
           java.lang.SecurityException
```

The `OperationControl` command enables the DAB client to change or read receiver parameters. The command is initiated by `operationControlReq` and is finalized by the confirmation `operationControlReq`.

Parameters:

`attribute` - The parameter can be set as follows:

- `DABConstants.operationControlSetVolume`: The volume of the receiver is set. The parameter value has to be of type `Integer` in the range from 0 to 100 (percent).
- `DABConstants.operationControlGetVolume`: The volume of the receiver is read. The parameter value is not considered.
- `DABConstants.operationControlSetServiceFollowing`: The service following feature is changed. Value has to be of type `Boolean`. If it is set to true and the receiver supports service following, then service following for services is switched on. If it is set to false, service following is switched off.
- `DABConstants.operationControlGetServiceFollowing`: Read the state of the service following. The parameter value is not considered.
- `DABConstants.operationControlGetServiceFollowingNotifications`: Instruct the package to send service following notifications. If the parameter value (of type `Boolean`) is set to true, the notifications are sent. If it is set to false, then no further notifications are sent.
- `DABConstants.operationControlSetDRCMODE`: Sets the DRC (Dynamic range control) mode. The DAB concept provides the option of Dynamic Range Control (DRC). The information is generated from the broadcaster's side (transported inside PAD, Programme Associated Data) to influence the audio output signal's dynamic range. The audio output signal will be modified if the option is activated with this call.

Value has to be of type `Boolean`. If it is set to true and the receiver supports DRC, then the DRC mode for audio services is switched on. If it is set to false, the DRC mode is switched off.

- `DABConstants.operationControlGetDRCMODE`: Read the state of the DRC mode. The parameter value is not considered.

- `DABConstants.operationControlGetDRCModeNotifications`: Instruct the package to send DRC mode notifications. If the parameter value (of type Boolean) is set to true, the notifications are sent. If it is set to false, then no further notifications are sent.

value - cf. description of attribute

See Also:

[operationControlCnf](#), [serviceFollowingNtf](#), [drcModeNtf](#)

Class ResourceConflictException

[dab](#)

```

java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--dab.DABException
            |
            +--dab.ResourceConflictException
  
```

All Implemented Interfaces:

java.io.Serializable

public Class **ResourceConflictException**

extends [DABException](#)

The exception indicates unsolved resource conflicts.

Version:

0.2

Methods inherited from class java.lang.Throwable

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Package dab.data

Interface Summary		Page
MOTObjectHeader	The MOTObjectHeader represents the header information of an MOT object Note: parameter values are defined in EN 301 234 "Digital Audio Broadcasting (DAB): Multimedia Object Transfer protocol"	358

Class Summary		Page
AnnouncementSupport	AnnouncementSupport represents supported announcement types of a certain DAB service, e.g.	384
BWSDirectoryIndex	The BWSDirectoryIndex class represents profile information in a BWS directory	383
BWSDirectoryObject	The BWSDirectoryObject class represents the carousel directory of a BWS user application.	382
BWSObject	The BWSObject class represents data that is part of the BWS service	379
DABObject	The DABObject class represents all kind of data that is transported via DAB.	379
DLSObject	The DLSObject represents data of the Dynamic Label Service.	375
Label	Label models a textual string which is used in the DAB System for service labels, object labels and so on.	374
LocationInfo	LocationInfo represents location data this is returned by the GetLocationInfo command.	372
MOTDirectoryObject	The MOTDirectoryObject class represents a MOT carousel directory of a component	370
MOTObject	The MOTObject represents data that is transported via the MOT protocol.	364
ObjectId	The ObjectId is an identifier for objects carried in a data service channel.	356
ProgrammeNumber	ProgrammeNumber represents a programme number that can be used for "programming" a service.	354
ProgrammeType	ProgrammeType represents provided programme types of a certain service.	353
SubscriberInfo	SubscriberInfo contains information how to subscribe to a service.	352

Class AnnouncementSupport

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+-+--dab.data.AnnouncementSupport
```

```
public Class AnnouncementSupport
```

```
extends java.lang.Object
```

AnnouncementSupport represents supported announcement types of a certain DAB service, e.g. News, Traffic and so on.

Version:

1.01

Constructor Summary		Page
AnnouncementSupport	(int announcementSupportFlags)	385

Method Summary		Page
boolean	equals (int announcementSupportFlags) Returns true when this object supports all the given flags; otherwise false	385
boolean	support (int announcement) Returns true when the announcement is supported; otherwise false	385

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

AnnouncementSupport

```
public AnnouncementSupport(int announcementSupportFlags)
```

Method Detail

equals

```
public boolean equals(int announcementSupportFlags)
```

Returns true when this object supports all the given flags; otherwise false

support

```
public boolean support(int announcement)
```

Returns true when the announcement is supported; otherwise false

Class **BWSDirectoryIndex**

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+-dab.data.BWSDirectoryIndex
```

```
public Class BWSDirectoryIndex
```

```
extends java.lang.Object
```

The `BWSDirectoryIndex` class represents profile information in a BWS directory

Version:

1.01

Constructor Summary		Page
	BWSDirectoryIndex ()	384

Method Summary		Page
java.lang.String	getIndexName () returns the index page name	384
int	getProfileId () returns the profile id that this index is for	384

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

BWSDirectoryIndex

```
public BWSDirectoryIndex()
```

Method Detail

getProfileId

```
public int getProfileId()
```

returns the profile id that this index is for

getIndexName

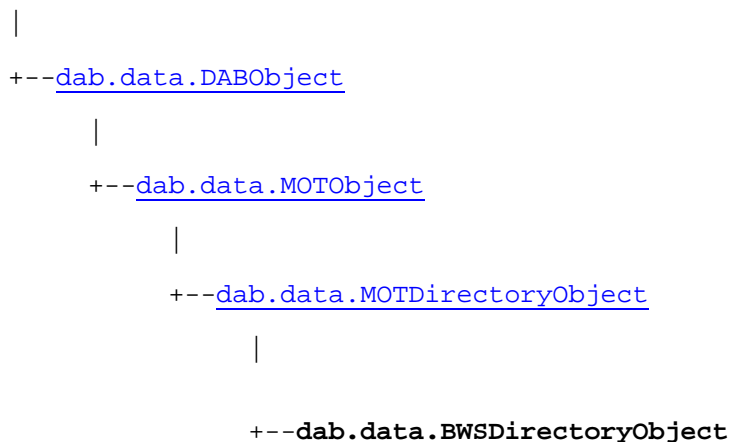
```
public java.lang.String getIndexName()
```

returns the index page name

Class *BWSDirectoryObject*

[dab.data](#)

java.lang.Object



All Implemented Interfaces:

[MOTObjectHeader](#)

public Class **BWSDirectoryObject**

extends [MOTDirectoryObject](#)

The *BWSDirectoryObject* class represents the carousel directory of a BWS user application.

Version:

1.01

Constructor Summary	Page
BWSDirectoryObject ()	383

Method Summary	Page
dab.data.BWSDirectoryIndex[] getDirectoryIndex () returns a list of profile index pages	383

Methods inherited from class dab.data. MOTDirectoryObject
getCarouselPeriod , getContents , getNumberOfObjects

Methods inherited from class dab.data. MOTObject
getBody , getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContentType , getCreationTime , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getValidity , getVersionNumber , toString

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface dab.data.[MOTObjectHeader](#)

[getCompressionType](#), [getContentDescription](#), [getContentDescriptionCharset](#), [getContentName](#), [getContentNameCharset](#), [getContentSubtype](#), [getContentSubtype](#), [getCreationTime](#), [getExpiration](#), [getExpireTime](#), [getLabel](#), [getMimeType](#), [getPriority](#), [getRepetitionDistance](#), [getStartValidity](#), [getTriggerTime](#), [getUniqueBodyVersion](#), [getValidity](#), [getVersionNumber](#)

Constructor Detail**BWSDirectoryObject**

```
public BWSDirectoryObject()
```

Method Detail**getDirectoryIndex**

```
public dab.data.BWSDirectoryIndex[] getDirectoryIndex()
```

returns a list of profile index pages

Class BWSObject[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.DABObject
```

```
|
```

```
+--dab.data.MOTObject
```

```
|
```

```
+--dab.data.BWSObject
```

All Implemented Interfaces:

[MOTObjectHeader](#)

```
public Class BWSObject
```

```
extends MOTObject
```

The BWSObject class represents data that is part of the BWS service

Version:

1.01

Constructor Summary		Page
BWSObject ()		381

Method Summary		Page
java.lang.String	getAdditionalHeader () Returns the additional header (the HTTP header field)	381
int	getCryptoAlgorithm () Returns the crypto algorithm for the object	381
byte[]	getProfileSubset () Returns the list of profiles for which the object is relevant	381
int	getScramblingMode () Returns the scrambling mode for the object	381
dab.data.SubscriberInfo	getSubscriberInfo () Returns information about how to subscribe to the service	381

Methods inherited from class dab.data. MOTOObject
getBody , getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContentType , getCreationTime , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getValidity , getVersionNumber , toString

Methods inherited from class java.lang.Object
equals , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Methods inherited from interface dab.data. MOTOObjectHeader
getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContentType , getCreationTime , getExpiration , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getUniqueBodyVersion , getValidity , getVersionNumber

Constructor Detail

BWSObject

```
public BWSObject()
```

Method Detail

getAdditionalHeader

```
public java.lang.String getAdditionalHeader ()
    throws DABNotAvailableException
```


Returns the additional header (the HTTP header field)

Throws:

[DABNotAvailableException](#) - when the content description is not available

getProfileSubset

```
public byte[] getProfileSubset()  
           throws DABNotAvailableException
```

Returns the list of profiles for which the object is relevant

Throws:

[DABNotAvailableException](#) - when the content description is not available

getCryptoAlgorithm

```
public int getCryptoAlgorithm()
```

Returns the crypto algorithm for the object

getScramblingMode

```
public int getScramblingMode()
```

Returns the scrambling mode for the object

getSubscriberInfo

```
public dab.data.SubscriberInfo getSubscriberInfo()  
           throws DABNotAvailableException
```

Returns information about how to subscribe to the service

Throws:

[DABNotAvailableException](#) - when the content description is not available

Class **DABObject**

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+-dab.data.DABObject
```

Direct Known Subclasses:

[DLSObject](#), [MOTOBJect](#)

```
abstract public Class DABObject
```

```
extends java.lang.Object
```

The **DABObject** class represents all kind of data that is transported via DAB.

Version:

1.01

Constructor Summary	Page
DABObject()	379

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail**DABObject**

```
public DABObject()
```

Class DLSObject[dab.data](#)

```
java.lang.Object
```

|

```
+--dab.data.DABObject
```

|

```
+--dab.data.DLSObject
```

```
public Class DLSObject
```

```
extends DABObject
```

The DLSObject represents data of the Dynamic Label Service.

Version:

1.02

Constructor Summary	Page
DLSObject()	377

Method Summary		Page
int	getCharacterFlagField() Returns the CharacterFlagField for the DLS	377
int	getCharSet() Returns the charSet of the DLS (cf.	377

java.lang.String	<u>getDynamicLabelSegment()</u> Returns the DLS converted to Unicode and without control characters Remark : Not all codetables for Unicode may be implemented on the receiver	377
int	<u>getEndofHeadlinePosition()</u> Returns the position of the last character belonging to the Headline inside the DLS	377
int[]	<u>getPreferedLineBreakPositions()</u> Returns the positions of the last character before a line break suggested by the broadcaster	378
int[]	<u>getPreferedWordBreakPositions()</u> Returns the positions of the last character before a word break suggested by the broadcaster	378
byte[]	<u>getRawDynamicLabelSegment()</u> Returns an array of bytes containing the DLS as it is.	377
int	<u>getSegmentNumber()</u> Returns the SegmentNumber	378
boolean	<u>isCommand()</u> Returns the Command Flag	378
boolean	<u>isToggle()</u> Returns the Toggle Flag	378

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

DLSObject

```
public DLSObject()
```

Method Detail

getRawDynamicLabelSegment

```
public byte[] getRawDynamicLabelSegment()
```

Returns an array of bytes containing the DLS as it is.

Remark : The CRC check for the DLS must be successfully passed

getDynamicLabelSegment

```
public java.lang.String getDynamicLabelSegment()
                        throws DABNotAvailableException
```

Returns the DLS converted to Unicode and without control characters

Remark : Not all codetables for Unicode may be implemented on the receiver

Throws:

[DABNotAvailableException](#) - when the information is not available

getCharSet

```
public int getCharSet()  
    throws DABNotAvailableException
```

Returns the charSet of the DLS (cf. DABConstants.charset*)

Throws:

[DABNotAvailableException](#) - when the information is not available

getCharacterFlagField

```
public int getCharacterFlagField()  
    throws DABNotAvailableException
```

Returns the CharacterFlagField for the DLS

Throws:

[DABNotAvailableException](#) - when the information is not available

getEndofHeadlinePosition

```
public int getEndofHeadlinePosition()  
    throws DABNotAvailableException
```

Returns the position of the last character belonging to the Headline inside the DLS

Throws:

[DABNotAvailableException](#) - when the information is not available

getPreferredLineBreakPositions

```
public int[] getPreferredLineBreakPositions()  
    throws DABNotAvailableException
```

Returns the positions of the last character before a line break suggested by the broadcaster

Throws:

[DABNotAvailableException](#) - when the information is not available

getPreferredWordBreakPositions

```
public int[] getPreferredWordBreakPositions()  
    throws DABNotAvailableException
```

Returns the positions of the last character before a word break suggested by the broadcaster

Throws:

[DABNotAvailableException](#) - when the information is not available

getSegmentNumber

```
public int getSegmentNumber()  
    throws DABNotAvailableException
```

Returns the SegmentNumber

Throws:

[DABNotAvailableException](#) - when the information is not available

isToggle

```
public boolean isToggle()
    throws DABNotAvailableException
```

Returns the Toggle Flag

Throws:

[DABNotAvailableException](#) - when the information is not available

isCommand

```
public boolean isCommand()
    throws DABNotAvailableException
```

Returns the Command Flag

Throws:

[DABNotAvailableException](#) - when the information is not available

Class Label

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.Label
```

```
public Class Label
```

```
extends java.lang.Object
```

Label models a textual string which is used in the DAB System for service labels, object labels and so on. It contains a text with max. 16 characters. Additionally the character set is indicated and it is specified how the label is to be displayed on a display with less than 16 characters.

Version:

1.01

Constructor Summary	Page
Label (int charSet, java.lang.String label, int characterFlagField)	375

Method Summary	Page
int getCharacterFlagField ()	377
Returns the character flag field	

int	getCharset() Returns the charset (cf.	375
java.lang.String	getLabel() Returns the content of the label	404

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Label

```
public Label(int charSet,
            java.lang.String label,
            int characterFlagField)
```

Method Detail

getCharset

```
public int getCharset()
```

Returns the charset (cf. DABConstants.charset*)

getCharacterFlagField

```
public int getCharacterFlagField()
```

Returns the character flag field

getLabel

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

Returns the content of the label

Class LocationInfo

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.LocationInfo
```

```
public Class LocationInfo
```

```
extends java.lang.Object
```

LocationInfo represents location data this is returned by the GetLocationInfo command.

Note, if the quality is below zero, than all other attributes are invalid.

The used coordinates have the same reference system as GPS.

Version:

1.01

Constructor Summary		Page
LocationInfo ()		373

Method Summary		Page
int	getAltitude () Returns the altitude in meters above ground.	373
int	getDirection () Returns the direction in 100.000ths of a degree (range: [0,360[in degrees; 0 degrees points to north).	374
int	getLatitude () Returns the latitude in 100.000ths of a degree (from +90 degrees for northerly latitudes to -90 degrees for southerly latitudes).	373
int	getLongitude () Returns the longitude in 100.000ths of a degree (from +180 degrees for easterly longitudes to -180 degrees for westerly longitudes).	373
int	getQuality () Returns the overall quality of the data.	374
int	getVelocity () Returns the velocity in 100.000ths of a meter per second.	374

Methods inherited from class java.lang.Object
<code>equals</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>

Constructor Detail**LocationInfo**

```
public LocationInfo()
```

Method Detail**getLongitude**

```
public int getLongitude()
```

Returns the longitude in 100.000ths of a degree (from +180 degrees for easterly longitudes to -180 degrees for westerly longitudes).

getLatitude

```
public int getLatitude()
```

Returns the latitude in 100.000ths of a degree (from +90 degrees for northerly latitudes to -90 degrees for southerly latitudes).

getAltitude

```
public int getAltitude()
```

Returns the altitude in meters above ground.

getVelocity

```
public int getVelocity()
```

Returns the velocity in 100.000ths of a meter per second.

getDirection

```
public int getDirection()
```

Returns the direction in 100.000ths of a degree (range: [0,360[in degrees; 0 degrees points to north).

getQuality

```
public int getQuality()
```

Returns the overall quality of the data. The range is from +100 (best) to -100 (worst). Negative values indicates invalid data.

Class MOTDirectoryObject

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.DABObject
```

```
|
```

```
+--dab.data.MOTObject
```

```
|
```

```
+--dab.data.MOTDirectoryObject
```

All Implemented Interfaces:

[MOTObjectHeader](#)

Direct Known Subclasses:

[BWSDirectoryObject](#)

public Class **MOTDirectoryObject**

extends [MOTOObject](#)

The MOTDirectoryObject class represents a MOT carousel directory of a component

Version:

1.02

Constructor Summary	Page
MOTDirectoryObject ()	371

Method Summary	Page
int getCarouselPeriod () get maximum time (in tenths of second) for the carousel to cycle	372
dab.data.MOTOObjectHeader[] getContents () Returns MOT Headers for objects described by the directory	372
int getNumberOfObjects () get number of objects described by the directory	371

Methods inherited from class dab.data. MOTOObject
getBody , getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContentType , getCreationTime , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getValidity , getVersionNumber , toString

Methods inherited from class java.lang.Object
equals , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Methods inherited from interface dab.data. MOTOObjectHeader
getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContentType , getCreationTime , getExpiration , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getUniqueBodyVersion , getValidity , getVersionNumber

Constructor Detail

MOTDirectoryObject

public **MOTDirectoryObject** ()

Method Detail**getNumberOfObjects**

```
public int getNumberOfObjects()
```

get number of objects described by the directory

Returns:

number of objects in the directory

getCarouselPeriod

```
public int getCarouselPeriod()
```

get maximum time (in tenths of second) for the carousel to cycle

Returns:

carousel period

getContents

```
public dab.data.MOTObjectHeader[] getContents()
```

Returns MOT Headers for objects described by the directory

Returns:

array of mot objects in the directory

Class MOTObject**dab.data**

```
java.lang.Object
```

```
|
```

```
+--dab.data.DABObject
```

```
|
```

```
+--dab.data.MOTObject
```

All Implemented Interfaces:

[MOTObjectHeader](#)

Direct Known Subclasses:

[BWSObject](#), [MOTDirectoryObject](#)

```
public Class MOTObject
```

```
extends DABObject
```

```
implements MOTObjectHeader
```

The MOTObject represents data that is transported via the MOT protocol.

Version:

1.07

Constructor Summary		Page
MOTObject ()		366

Method Summary		Page
byte[]	getBody () Returns the body of the object (the actual content)	366
int	getCompressionType () Returns the compression type of the object	370
java.lang.String	getContentDescription () Returns the content description	366
int	getContentDescriptionCharset () Returns the charset of the content description (cf.	367
java.lang.String	getContentName () Returns the content name	367
int	getContentNameCharset () Returns the charset of the content name (cf.	367
int	getContentSubtype () Returns the content subtype (the exact type)	366
int	getContentType () Returns the content type (the main category)	366
java.util.Date	getCreationTime () Returns the authoring date of the object	368
java.util.Date	getExpireTime () Returns the date after which the object is not valid anymore	369
dab.data.Label	getLabel () Returns the label	404
java.lang.String	getMimeType () Returns the MIME type of the object	370
int	getPriority () Returns the priority (0=lowest priority; 255=highest priority)	368
int	getRepetitionDistance () Returns the repetition distance (in ms)	368
java.util.Date	getStartValidity () Returns the date after which the object is valid	369
java.util.Date	getTriggerTime () Returns the date for presenting the object	369

boolean	getValidity() Returns false, if validity is now; otherwise true.	368
int	getVersionNumber() Returns the version number of the object	368
Java.lang.String	toString() Returns a textual representation of the object	389

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface dab.data.[MOTObjectHeader](#)

[getCompressionType](#), [getContentDescription](#), [getContentDescriptionCharset](#), [getContentName](#), [getContentNameCharset](#), [getContentSubtype](#), [getContentType](#), [getCreationTime](#), [getExpiration](#), [getExpireTime](#), [getLabel](#), [getMimeType](#), [getPriority](#), [getRepetitionDistance](#), [getStartValidity](#), [getTriggerTime](#), [getUniqueBodyVersion](#), [getValidity](#), [getVersionNumber](#)

Constructor Detail

MOTObject

```
public MOTObject()
```

Method Detail

getContentType

```
public int getContentType()
```

Returns the content type (the main category)

Specified by:

[getContentType](#) in interface [MOTObjectHeader](#)

getContentSubtype

```
public int getContentSubtype()
```

Returns the content subtype (the exact type)

Specified by:

[getContentSubtype](#) in interface [MOTObjectHeader](#)

getBody

```
public byte[] getBody()
```

Returns the body of the object (the actual content)

getContentDescription

```
public java.lang.String getContentDescription()  
                        throws DABNotAvailableException
```

Returns the content description

Specified by:

[getContentDescription](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the content description is not available

getContentDescriptionCharset

```
public int getContentDescriptionCharset()  
          throws DABNotAvailableException
```

Returns the charset of the content description (cf. DABConstants.charset*)

Specified by:

[getContentDescriptionCharset](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the charset is not available

getContentName

```
public java.lang.String getContentName()  
                       throws DABNotAvailableException
```

Returns the content name

Specified by:

[getContentName](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the content name is not available

getContentNameCharset

```
public int getContentNameCharset()  
          throws DABNotAvailableException
```

Returns the charset of the content name (cf. DABConstants.charset*)

Specified by:

[getContentNameCharset](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the charset is not available

getLabel

```
public dab.data.Label getLabel()  
                    throws DABNotAvailableException
```

Returns the label

Specified by:

[getLabel](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the priority is not available

getPriority

```
public int getPriority()  
    throws DABNotAvailableException
```

Returns the priority (0=lowest priority; 255=highest priority)

Specified by:

[getPriority](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the priority is not available

getRepetitionDistance

```
public int getRepetitionDistance()  
    throws DABNotAvailableException
```

Returns the repetition distance (in ms)

Specified by:

[getRepetitionDistance](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the repetition distance is not available

getVersionNumber

```
public int getVersionNumber()  
    throws DABNotAvailableException
```

Returns the version number of the object

Specified by:

[getVersionNumber](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the version is not available

getValidity

```
public boolean getValidity()
```

Returns false, if validity is now; otherwise true. Note, if the validity is set to false the referred time routines have to be ignored.

Specified by:

[getValidity](#) in interface [MOTObjectHeader](#)

See Also:

[getCreationTime](#), [getStartValidity](#), [getExpireTime](#), [getTriggerTime](#)

getCreationTime

```
public java.util.Date getCreationTime()  
                        throws DABNotAvailableException
```

Returns the authoring date of the object

Specified by:

[getCreationTime](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getStartValidity

```
public java.util.Date getStartValidity()  
                        throws DABNotAvailableException
```

Returns the date after which the object is valid

Specified by:

[getStartValidity](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getExpireTime

```
public java.util.Date getExpireTime()  
                        throws DABNotAvailableException
```

Returns the date after which the object is not valid anymore

Specified by:

[getExpireTime](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getTriggerTime

```
public java.util.Date getTriggerTime()  
                        throws DABNotAvailableException
```

Returns the date for presenting the object

Specified by:

[getTriggerTime](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when not available

See Also:[getValidity](#)

toString

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

Returns a textual representation of the object

Overrides:

toString in class java.lang.Object

getMimeType

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

Returns the MIME type of the object

Specified by:

[getMimeType](#) in interface [MOTObjectHeader](#)

getCompressionType

```
public int getCompressionType()  
        throws DABNotAvailableException
```

Returns the compression type of the object

Specified by:

[getCompressionType](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the content description is not available

Interface MOTObjectHeader

[dab.data](#)

All Known Implementing Classes:

[BWSDirectoryObject](#), [BWSObject](#), [MOTDirectoryObject](#), [MOTObject](#)

```
public Interface MOTObjectHeader
```

The MOTObjectHeader represents the header information of an MOT object

Note: parameter values are defined in EN 301 234 "Digital Audio Broadcasting (DAB): Multimedia Object Transfer protocol"

Version:

1.03

Method Summary		Page
int	getCompressionType () get the compression type of the object	370
java.lang.String	getContentDescription () Deprecated. <i>obsolete parameter as defined in EN 301 234</i>	366
int	getContentDescriptionCharset () Deprecated. <i>obsolete parameter as defined in EN 301 234</i>	367
java.lang.String	getContentName () get the content name	367
int	getContentNameCharset () get the charset of the content name (see DABConstants.charset*)	367
int	getContentSubtype () get the content subtype (the exact type) as defined in EN 301 234	366
int	getContentType () get the content type (the main category) as defined in EN 301 234	366
java.util.Date	getCreationTime () Deprecated. <i>obsolete parameter as defined in EN 301 234</i>	368
int	getExpiration () get the relative expiration time in minutes - if an absolute expire time is signalled expiration has to be transformed to a value in minutes - if an relative expire time is signalled the expiration is always be transformed in a value in minutes	363
java.util.Date	getExpireTime () Deprecated. <i>obsolete parameter as defined in EN 301 234, use getExpiration() instead</i>	369
dab.data.Label	getLabel () get the label	404
java.lang.String	getMimeType () Returns the MIME type of the object	370
int	getPriority () get the priority (0=lowest priority; 255=highest priority) NOTE: for MOT directory mode only	368
int	getRepetitionDistance () get the repetition distance (in 1/10s)	368
java.util.Date	getStartValidity () Deprecated. <i>obsolete parameter as defined in EN 301 234</i>	369
java.util.Date	getTriggerTime () the date for presenting the object	369
int	getUniqueBodyVersion () get the unique body version of the object	362

boolean	getValidity() returns false, if validity is now; otherwise true.	368
int	getVersionNumber() Deprecated. <i>obsolete parameter as defined in EN 301 234, use getUniqueBodyVersion() instead</i>	368

Method Detail

getContent-type

```
public int getContent-type()
```

get the content type (the main category) as defined in EN 301 234

Returns:

MOT Content-type

getContentSubtype

```
public int getContentSubtype()
```

get the content subtype (the exact type) as defined in EN 301 234

Returns:

MOT ContentSubType

getContentDescription

```
public java.lang.String getContentDescription()  
throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234*

get the content description

Returns:

MOT ContentDescription

Throws:

[DABNotAvailableException](#) - when the content description is not available

getContentDescriptionCharset

```
public int getContentDescriptionCharset()  
throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234*

get the charset of the content description (see `DABConstants.charset*`)

Returns:

MOT charsetset of ContentDescription

Throws:

[DABNotAvailableException](#) - when the charset is not available

See Also:

DABConstants

getContentName

```
public java.lang.String getContentName()  
    throws DABNotAvailableException
```

get the content name

Returns:

MOT ContentName

Throws:

[DABNotAvailableException](#) - when the content name is not available

getContentNameCharset

```
public int getContentNameCharset()  
    throws DABNotAvailableException
```

get the charset of the content name (see DABConstants.charset*)

Returns:

MOT ContentName Characterset

Throws:

[DABNotAvailableException](#) - when the charset is not available

See Also:

DABConstants

getLabel

```
public dab.data.Label getLabel()  
    throws DABNotAvailableException
```

get the label

Returns:

MOT Label

Throws:

[DABNotAvailableException](#) - when the label is not available

getPriority

```
public int getPriority()  
    throws DABNotAvailableException
```

get the priority (0=lowest priority; 255=highest priority)

NOTE: for MOT directory mode only

Returns:

MOT Priority as defined in as defined in EN 301 234

Throws:

[DABNotAvailableException](#) - when the priority is not available

getRepetitionDistance

```
public int getRepetitionDistance()  
    throws DABNotAvailableException
```

get the repetition distance (in 1/10s)

Returns:

MOT repetition distance in 1/10s

Throws:

[DABNotAvailableException](#) - when the repetition distance is not available

getVersionNumber

```
public int getVersionNumber()  
        throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234, use `getUniqueBodyVersion()` instead*

get the version number of the object

Returns:

MOT version number of object

Throws:

[DABNotAvailableException](#) - when the version number is not available

See Also:

[getUniqueBodyVersion](#)

getUniqueBodyVersion

```
public int getUniqueBodyVersion()  
        throws DABNotAvailableException
```

get the unique body version of the object

Returns:

MOT unique body version as defined in EN 301 234

Throws:

[DABNotAvailableException](#) - when the unique body version is not available

Since:

version 1.03

getValidity

```
public boolean getValidity()
```

returns false, if validity is now; otherwise true. Note, if the validity is set to false the referred time routines have to be ignored.

See Also:

[getCreationTime](#), [getStartValidity](#), [getExpireTime](#), [getTriggerTime](#)

getCreationTime

```
public java.util.Date getCreationTime()  
        throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234*

get the authoring date of the object

Returns:

the authoring date of the object

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getStartValidity

```
public java.util.Date getStartValidity()  
        throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234*

get the date after which the object is valid

Returns:

date after which the object is valid

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getExpireTime

```
public java.util.Date getExpireTime()  
    throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234, use getExpiration() instead*

get the date after which the object is not valid anymore

Returns:

the date after which the object is not valid anymore

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#), [getExpiration](#)

getExpiration

```
public int getExpiration()  
    throws DABNotAvailableException
```

get the relative expiration time in minutes

- if an absolute expire time is signalled expiration has to be transformed to a value in minutes - if an relative expire time is signalled the expiration is always be transformed in a value in minutes

Returns:

the relative expiration time in minutes

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getTriggerTime

```
public java.util.Date getTriggerTime()  
    throws DABNotAvailableException
```

the date for presenting the object

Returns:

the date for presenting the object

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getMimeType

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

Returns the MIME type of the object

getCompressionType

```
public int getCompressionType()
    throws DABNotAvailableException
```

get the compression type of the object

Returns:

the compression type of the object as defined in EN 301 234

Throws:

[DABNotAvailableException](#) - when the content description is not available

Class ObjectId

dab.data

```
java.lang.Object
```

```
|
```

```
+--dab.data.ObjectId
```

```
public Class ObjectId
```

```
extends java.lang.Object
```

The ObjectId is an identifier for objects carried in a data service channel. It is used to request objects and for identification of delivered objects to the application.

Version:

1.04

Constructor Summary	Page
ObjectId () Constructs an ObjectId object	357
ObjectId (dab.data.ObjectId objectId) Constructs a copy of the given ObjectId.	357
ObjectId (java.lang.String stringId) Constructs an ObjectId object from the given string.	357

Method Summary	Page
int compareTo (dab.data.ObjectId objectId) This method compares the object with the given object.	358

int	compareTo (java.lang.Object objectId) This method compares the object with the given object.	391
java.lang.String	getId () Returns an external representation of the identifier in a textual format.	403

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

ObjectId

```
public ObjectId()
```

Constructs an ObjectId object

ObjectId

```
public ObjectId(dab.data.ObjectId objectId)
```

Constructs a copy of the given ObjectId.

See Also:

[getId](#)

ObjectId

```
public ObjectId(java.lang.String stringId)
```

Constructs an ObjectId object from the given string.

See Also:

[getId](#)

Method Detail

compareTo

```
public int compareTo(java.lang.Object objectId)
```

This method compares the object with the given object. The behavior is the same as it is specified in the `compareTo` method of the `java.lang.Comparable` interface.

compareTo

```
public int compareTo(dab.data.ObjectId objectId)
```

This method compares the object with the given object. The behavior is the same as it is specified in the `compareTo` method of the `java.lang.Comparable` interface.

getId

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

Returns an external representation of the identifier in a textual format. The returned value can be used to construct an object id.

Class ProgrammeNumber

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.ProgrammeNumber
```

```
public Class ProgrammeNumber
```

```
extends java.lang.Object
```

ProgrammeNumber represents a programme number that can be used for "programming" a service.

Version:

1.00

Constructor Summary	<i>Page</i>
ProgrammeNumber ()	355

Method Summary		<i>Page</i>
dab.si.ServiceId	getNewService () returns the ServiceId of the new service when the programme is redirected	356
java.util.Date	getTransmissionTime () returns the transmission time	355
boolean	isInterrupted () signals, whether the programme is interrupted by later continued	356
boolean	isRedirected () signals, whether the programme is redirected to a different service and time	356

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ProgrammeNumber

```
public ProgrammeNumber()
```

Method Detail

getTransmissionTime

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

returns the transmission time

isInterrupted

```
public boolean isInterrupted()
```

signals, whether the programme is interrupted by later continued

isRedirected

```
public boolean isRedirected()
```

signals, whether the programme is redirected to a different service and time

See Also:

[getNewService](#)

getNewService

```
public dab.si.ServiceId getNewService()
```

returns the ServiceId of the new service when the programme is redirected

See Also:

[isRedirected](#)

Class ProgrammeType

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.ProgrammeType
```

```
public Class ProgrammeType
```

```
extends java.lang.Object
```

ProgrammeType represents provided programme types of a certain service. It consists of an international code, an optional coarse code and two optional fine codes.

Version:

1.01

Constructor Summary		Page
ProgrammeType (int internationalCode, byte[] fineCode, boolean hasCoarseCode, int coarseCode)		354

Method Summary		Page
int getCoarseCode ()	Returns the coarse code	354
byte[] getFineCode ()	Returns a reference to the fine codes	354
int getInternationalCode ()	Returns the international code	354

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

ProgrammeType

```
public ProgrammeType(int internationalCode,
                    byte[] fineCode,
                    boolean hasCoarseCode,
                    int coarseCode)
```

Method Detail

getInternationalCode

```
public int getInternationalCode()
```

Returns the international code

getCoarseCode

```
public int getCoarseCode()
    throws DABNotAvailableException
```

Returns the coarse code

Throws:

[DABNotAvailableException](#) - when the code is not available

getFineCode

```
public byte[] getFineCode()
```

Returns a reference to the fine codes

Class SubscriberInfo

[dab.data](#)

java.lang.Object

|

+-**dab.data.SubscriberInfo**

public Class **SubscriberInfo**

extends java.lang.Object

SubscriberInfo contains information how to subscribe to a service.

Version:

1.00

Constructor Summary	Page
SubscriberInfo ()	352

Method Summary	Page
java.lang.String getContentName () returns the content name of the alternative object	367
int getEncryptionSpecificFlags () returns a flag field that can be used for redirection purposes	353
int getReason () returns a flag field (cf.	452

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SubscriberInfo

public **SubscriberInfo**()

Method Detail

getReason

public int **getReason**()

returns a flag field (cf. DABConstants.subscriberInfo*) that explains why the related BWS object could not be descrambled

getEncryptionSpecificFlags

```
public int getEncryptionSpecificFlags()
```

returns a flag field that can be used for redirection purposes

getContentName

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

returns the content name of the alternative object

Package **dab.events**

Class Summary		<i>Page</i>
AppStateChangeEvent	AppStateChangeEvent reflects state changes in an application.	452
ComponentNtfEvent		451
ConflictResolutionNtfEvent		449
DABEvent	DABEvent is the superclass for all events used inside the DAB package.	448
DRCMModeNtfEvent	The DRCModeNtfEvent is generated when a DRC mode change is taken by the receiver.	447
GetComponentInfoCnfEvent	The GetComponentInfoCnfEvent is generated in response to a GetComponentInfoReq request.	446
GetEnsembleInfoCnfEvent	The GetEnsembleInfoCnfEvent is generated in response to a GetEnsembleInfoReq request.	445
GetLocationInfoCnfEvent	The GetLocationInfoCnfEvent is generated in response to a getLocationInfoReq request.	443
GetServiceInfoCnfEvent	The GetServiceInfoCnfEvent is generated in response to a GetServiceInfoReq request.	441
LocationInfoNtfEvent	The LocationInfoNtfEvent represents notifications related to the GetLocationInfo command.	440
ObjectNtfEvent	The ObjectNtfEvent is generated in response to a ObjectNtfReq request.	438
OperationControlCnfEvent	OperationControlCnfEvent is generated in response to a operationControlReq request.	436
ReceptionInfoNtfEvent	ReceptionInfoNtfEvent is generated in response to a selectReceptionInfoReq request.	434
RespondConflictResolutionCnfEvent		433
ScanCnfEvent	ScanCnfEvent is generated in response to a scanReq request.	431
ScanNtfEvent	ScanNtfEvent is generated in response to a scanReq request.	429
SearchCnfEvent	SearchCnfEvent is generated in response to a searchReq request.	427
SearchNtfEvent	SearchNtfEvent is generated in response to a searchReq request.	425
SelectApplicationCnfEvent	SelectApplicationCnfEvent is generated in response to a selectApplicationReq request.	424

SelectComponentCnfEvent	The SelectComponentCnfEvent is generated in response to a selectComponentReq request.	422
SelectComponentStreamCnfEvent	SelectComponentStreamCnfEvent is generated in response to a selectComponentStreamReq.	421
SelectObjectCnfEvent	The SelectObjectCnfEvent is generated in response to a selectObjectReq request.	419
SelectReceptionInfoCnfEvent	The SelectReceptionInfoCnfEvent is generated in response to a selectReceptionInfoReq request.	417
SelectSICnfEvent	The SelectSICnfEvent is generated in response to a selectSIReq request.	414
ServiceFollowingNtfEvent	The ServiceFollowingNtfEvent is generated when a service following action is taken by the receiver.	413
SINtfEvent	The SINtfEvent is generated in response to a selectSIReq request.	409
SystemFailureNtfEvent	The System-Failure event is generated when a fatal error has happened in the system.	408
TuneCnfEvent	The TuneCnfEvent is generated in response to a tuneReq request.	406

Class AppStateChangeEvent

[dab.events](#)

java.lang.Object

|

+- -dab.events.AppStateChangeEvent

public Class **AppStateChangeEvent**

extends java.lang.Object

AppStateChangeEvent reflects state changes in an application. Note: to support J2ME MIDP profile AppStateChangeEvent does not extend java.util.EventObject anymore

Version:

0.3

See Also:

[AppStateChangeListener](#), "Digital Video Broadcasting (DVB) Multimedia Home Platform (MHP), ETSI TS 101 812"

Constructor Summary		Page
protected	AppStateChangeEvent (dab.AppStateChangeEventSource source, int fromState, int toState, boolean failed)	453

Method Summary		Page
int	getFromState () Returns the state from which the application was switching	453

int	getToState() Returns the state to which the application switched	453
boolean	hasFailed() Indicates whether the switching failed (=true) or not (=false)	454

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

AppStateChangeEvent

```
protected AppStateChangeEvent(dab.AppStateChangeEventSource source,
                               int fromState,
                               int toState,
                               boolean failed)
```

Method Detail

getFromState

```
public int getFromState()
```

Returns the state from which the application was switching

getToState

```
public int getToState()
```

Returns the state to which the application switched

hasFailed

```
public boolean hasFailed()
```

Indicates whether the switching failed (=true) or not (=false)

Class ComponentNtfEvent

[dab.events](#)

`java.lang.Object`

|

+--[dab.events.DABEvent](#)

|

+--`dab.events.ComponentNtfEvent`

public Class **ComponentNtfEvent**

extends [DABEvent](#)

Version:

1.02

Constructor Summary		Page
protected	ComponentNtfEvent (dab.DABSource source, int reason, dab.si.ComponentId componentId, int selectionMode)	452

Method Summary		Page
dab.si.ComponentId	getComponentId () Returns the component which is involved	452
int	getReason () Returns the reason for change of the selectionMode (the code is compatible with DABConstants.result*).	452
int	getSelectionMode () Returns the new selection mode for the component.	452

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

ComponentNtfEvent

```
protected ComponentNtfEvent(dab.DABSource source,
                             int reason,
                             dab.si.ComponentId componentId,
                             int selectionMode)
```

Method Detail

getReason

```
public int getReason()
```

Returns the reason for change of the selectionMode (the code is compatible with DABConstants.result*).

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the component which is involved

getSelectionMode

```
public int getSelectionMode()
```

Returns the new selection mode for the component.

See Also:

[getSelectionMode](#)

Class ConflictResolutionNtfEvent

[dab.events](#)

```
java.lang.Object
```

```

|
+--dab.events.DABEvent
|
+--dab.events.ConflictResolutionNtfEvent

```

```
public Class ConflictResolutionNtfEvent
```

```
extends DABEvent
```

Constructor Summary		Page
protected	ConflictResolutionNtfEvent (dab.DABSource source, int _transaction, int _turn, int _operation, int _suboperation)	450

Method Summary		Page
int	getOperation () Gives back a code of the involved operation (cf.	451
int	getSuboperation () Gives back a code of the involved suboperation (cf.	451
int	getTransaction () Delivers the transaction number.	450
int	getTurn () Returns a code for the turn of the resource conflict resolution protocol: <ul style="list-style-type: none"> • DABConstants.conflictResolutionTurnProceed: This is sent to the client which requested the operation. 	450

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```


Constructor Detail

ConflictResolutionNtfEvent

```
protected ConflictResolutionNtfEvent(dab.DABSource source,  
                                     int _transaction,  
                                     int _turn,  
                                     int _operation,  
                                     int _suboperation)
```

Method Detail

getTransaction

```
public int getTransaction()
```

Delivers the transaction number. This can be used to provide a transaction context.

getTurn

```
public int getTurn()
```

Returns a code for the turn of the resource conflict resolution protocol:

- `DABConstants.conflictResolutionTurnProceed`: This is sent to the client which requested the operation. It indicates that there is a resource conflict. The client is asked whether he likes to proceed.
- `DABConstants.conflictResolutionTurnProbe`: This notification is sent to all clients in order to probe for their willingness to release the needed resources.
- `conflictResolutionTurnStop`: The client is asked to stop the indicated operation in order to release the resources.
- `DABConstants.conflictResolutionTurnPreempt`: The client is informed that the indicated operation was stopped. This action shall normally only be taken, when the client failed to do a stop in the previous turn.

getOperation

```
public int getOperation()
```

Gives back a code of the involved operation (cf. `DABConstants.conflictResolutionOperation*`)

getSuboperation

```
public int getSuboperation()
```

Gives back a code of the involved suboperation (cf. `DABConstants.conflictResolutionSuboperation*`)

Class **DABEvent**

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

Direct Known Subclasses:

[ComponentNtfEvent](#), [ConflictResolutionNtfEvent](#), [DRCModeNtfEvent](#), [GetComponentInfoCnfEvent](#), [GetEnsembleInfoCnfEvent](#), [GetLocationInfoCnfEvent](#), [GetServiceInfoCnfEvent](#), [LocationInfoNtfEvent](#), [ObjectNtfEvent](#), [OperationControlCnfEvent](#), [ReceptionInfoNtfEvent](#), [RespondConflictResolutionCnfEvent](#), [ScanCnfEvent](#), [ScanNtfEvent](#), [SearchCnfEvent](#), [SearchNtfEvent](#), [SelectApplicationCnfEvent](#), [SelectComponentCnfEvent](#), [SelectComponentStreamCnfEvent](#), [SelectObjectCnfEvent](#), [SelectReceptionInfoCnfEvent](#), [SelectSICnfEvent](#), [ServiceFollowingNtfEvent](#), [SINtfEvent](#), [SystemFailureNtfEvent](#), [TuneCnfEvent](#)

public Class **DABEvent**

extends java.lang.Object

DABEvent is the superclass for all events used inside the DAB package. Note: to support J2ME MIDP profile DABEvent does not extend java.util.EventObject anymore

Version:

1.02

Constructor Summary		Page
protected	DABEvent (dab.DABSource source)	449

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail**DABEvent**

protected **DABEvent**(dab.DABSource source)

Class DRCModeNtfEvent

[dab.events](#)

java.lang.Object

|

+--[dab.events.DABEvent](#)

|

+--**dab.events.DRCModeNtfEvent**

public Class **DRCModeNtfEvent**

extends [DABEvent](#)

The DRCModeNtfEvent is generated when a DRC mode change is taken by the receiver.

Version:

1.02

See Also:[serviceFollowingNtf](#)

Constructor Summary		Page
protected	DRCModeNtfEvent (dab.DABSource source, boolean currentState)	448
	Creates an DRCModeNtfEvent event.	

Method Summary		Page
boolean	getMode ()	448
	Returns the DRC mode.	

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail**DRCModeNtfEvent**

```
protected DRCModeNtfEvent(dab.DABSource source,
                          boolean currentState)
```

Creates an DRCModeNtfEvent event.

Method Detail**getMode**

```
public boolean getMode()
```

Returns the DRC mode. The return value is true, if it is now switched on; it is false when it is now switched off.

Class GetComponentInfoCnfEvent[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.GetComponentInfoCnfEvent
```

```
public Class GetComponentInfoCnfEvent
```

```
extends DABEvent
```

The *GetComponentInfoCnfEvent* is generated in response to a *GetComponentInfoReq* request.

Version:

1.01

See Also:

[GetComponentInfoCnf](#)

Constructor Summary		Page
protected	GetComponentInfoCnfEvent (dab.DABSource source, int result, dab.si.ComponentInfo componentInfo) Create a <i>GetComponentInfoCnfEvent</i> object.	447

Method Summary		Page
dab.si.ComponentInfo	GetComponentInfo () Returns information about the subscribed DAB Component.	447
int	getResult () Returns the status of the <i>GetComponentInfo</i> command.	447

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

GetComponentInfoCnfEvent

```
protected GetComponentInfoCnfEvent(dab.DABSource source,
                                     int result,
                                     dab.si.ComponentInfo componentInfo)
```

Create a *GetComponentInfoCnfEvent* object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the *GetComponentInfo* command. If it is equal to *DABConstants.resultOK*, the command was successful. Otherwise an error has occurred.

GetComponentInfo

```
public dab.si.ComponentInfo GetComponentInfo()
```

Returns information about the subscribed DAB Component.

Class *GetEnsembleInfoCnfEvent*

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.GetEnsembleInfoCnfEvent

```

public Class **GetEnsembleInfoCnfEvent**

extends [DABEvent](#)

The *GetEnsembleInfoCnfEvent* is generated in response to a *GetEnsembleInfoReq* request.

Version:

1.01

See Also:

[getEnsembleInfoCnf](#)

Constructor Summary		Page
protected	GetEnsembleInfoCnfEvent (dab.DABSource source, int result, dab.si.EnsembleInfo ensembleInfo) Creates a <i>GetEnsembleInfoCnfEvent</i> object.	445

Method Summary		Page
dab.si.EnsembleInfo	getEnsembleInfo () Returns a reference to an object which provides information about a DAB Ensemble.	446
int	getResult () Returns the status of the <i>GetEnsembleInfo</i> command.	447

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

GetEnsembleInfoCnfEvent

```

protected GetEnsembleInfoCnfEvent(dab.DABSource source,
                                     int result,
                                     dab.si.EnsembleInfo ensembleInfo)

```

Creates a *GetEnsembleInfoCnfEvent* object.

Method Detail**getResult**

```
public int getResult()
```

Returns the status of the GetEnsembleInfo command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getEnsembleInfo

```
public dab.si.EnsembleInfo getEnsembleInfo()
```

Returns a reference to an object which provides information about a DAB Ensemble.

Class GetLocationInfoCnfEvent[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.GetLocationInfoCnfEvent
```

```
public Class GetLocationInfoCnfEvent
```

```
extends DABEvent
```

The GetLocationInfoCnfEvent is generated in response to a getLocationInfoReq request.

Version:

1.01

See Also:

[getLocationInfoCnf](#)

Constructor Summary		Page
protected	GetLocationInfoCnfEvent (dab.DABSource source, int result, int mode, int deliveredDelta, int deliveredAccuracy) Creates a GetLocationInfoCnfEvent object.	444

Method Summary		Page
int	getDeliveredAccuracy () Returns the delivered accuracy of the GetLocationInfo command.	444

int	getDeliveredDelta() Returns the delivered delta of the GetLocationInfo command.	444
int	getMode() Returns the mode of the GetLocationInfo command.	448
int	getResult() Returns the status of the GetLocationInfo command.	447

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

GetLocationInfoCnfEvent

```
protected GetLocationInfoCnfEvent(dab.DABSource source,
                                     int result,
                                     int mode,
                                     int deliveredDelta,
                                     int deliveredAccuracy)
```

Creates a GetLocationInfoCnfEvent object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the GetLocationInfo command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getMode

```
public int getMode()
```

Returns the mode of the GetLocationInfo command.

See Also:

[getLocationInfoReq](#)

getDeliveredDelta

```
public int getDeliveredDelta()
```

Returns the delivered delta of the GetLocationInfo command.

See Also:

[getLocationInfoReq](#)

getDeliveredAccuracy

```
public int getDeliveredAccuracy()
```

Returns the delivered accuracy of the GetLocationInfo command.

See Also:

[getLocationInfoReq](#)

Class **GetServiceInfoCnfEvent**

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.GetServiceInfoCnfEvent

```

public Class **GetServiceInfoCnfEvent**

extends [DABEvent](#)

The GetServiceInfoCnfEvent is generated in response to a GetServiceInfoReq request.

Version:

1.01

See Also:

[getServiceInfoCnf](#)

Constructor Summary		Page
protected	GetServiceInfoCnfEvent (dab.DABSource source, int result, dab.si.ServiceInfo serviceInfo) Creates a GetServiceInfoCnfEvent object.	442

Method Summary		Page
int	getResult () Returns the status of the GetServiceInfo command.	447
dab.si.ServiceInfo	getServiceInfo () Returns a reference to an object which provides information about the DAB Service.	442

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail**GetServiceInfoCnfEvent**

```
protected GetServiceInfoCnfEvent(dab.DABSource source,
                                   int result,
                                   dab.si.ServiceInfo serviceInfo)
```

Creates a GetServiceInfoCnfEvent object.

Method Detail**getResult**

```
public int getResult()
```

Returns the status of the GetServiceInfo command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getServiceInfo

```
public dab.si.ServiceInfo getServiceInfo()
```

Returns a reference to an object which provides information about the DAB Service.

Class LocationInfoNtfEvent[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.LocationInfoNtfEvent
```

```
public Class LocationInfoNtfEvent
```

```
extends DABEvent
```

The LocationInfoNtfEvent represents notifications related to the GetLocationInfo command.

Version:

1.02

See Also:

[locationInfoNtf](#)

Constructor Summary		Page
protected	LocationInfoNtfEvent (dab.DABSource source, java.util.Date timestamp, int[] regionIds, dab.data.LocationInfo info) Creates a LocationInfoNtfEvent object.	441

Method Summary		Page
dab.data.LocationInfo	getLocationInfo () Returns the location info.	441
int[]	getRegionIds () Returns the list of region identifiers.	441
java.util.Date	getTimestamp () Returns the timestamp	441

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

LocationInfoNtfEvent

```
protected LocationInfoNtfEvent(dab.DABSource source,
                                java.util.Date timestamp,
                                int[] regionIds,
                                dab.data.LocationInfo info)
```

Creates a LocationInfoNtfEvent object.

Method Detail

getTimestamp

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

Returns the timestamp

getRegionIds

```
public int[] getRegionIds()
```

Returns the list of region identifiers. When no region ids are available or are not requested, the result is an empty array.

getLocationInfo

```
public dab.data.LocationInfo getLocationInfo()
```

Returns the location info. When the location info was not requested, the result is null.

Class **ObjectNtfEvent**

[dab.events](#)

java.lang.Object

|

+--[dab.events.DABEvent](#)

|

+--**dab.events.ObjectNtfEvent**

public Class **ObjectNtfEvent**

extends [DABEvent](#)

The ObjectNtfEvent is generated in response to a ObjectNtfReq request.

Version:

1.02

See Also:

[objectNtf](#)

Constructor Summary		Page
protected	ObjectNtfEvent (dab.DABSource source, dab.si.ComponentId componentId, dab.data.ObjectId objectId, int selectionState, dab.data.DABObject object) Creates an ObjectNtfEvent event.	439

Method Summary		Page
dab.si.ComponentId	getComponentId () Returns the component the object is belonging to.	452
dab.data.DABObject	getObject () Returns a reference to a DAB object.	440
dab.data.ObjectId	getObjectId () Returns the id of the selected object.	439
int	getSelectionState () Returns the current selection state.	439

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ObjectNtfEvent

```
protected ObjectNtfEvent(dab.DABSource source,  
                        dab.si.ComponentId componentId,  
                        dab.data.ObjectId objectId,  
                        int selectionState,  
                        dab.data.DABObject object)
```

Creates an ObjectNtfEvent event.

Method Detail

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the component the object is belonging to.

getObjectId

```
public dab.data.ObjectId getObjectId()
```

Returns the id of the selected object.

getSelectionState

```
public int getSelectionState()
```

Returns the current selection state. The following values are supported:

- DABConstants.selectionStateOK: This message delivers a selected object to the connected DAB client. The object is available by parameter object.
- DABConstants.selectionStateDelayed: Delivery of the selected object is delayed.
- DABConstants.selectionStateTerminated: Transmission of the selected object is terminated. The object selection is removed.

getObject

```
public dab.data.DABObject getObject()
```

Returns a reference to a DAB object. As the DABObject class is just an abstraction of data objects, you have to check the actual type of the returned object (e.g. instanceof MOTObject) to know which kind of object is delivered.

Class *OperationControlCnfEvent*

[dab.events](#)

java.lang.Object

|

+--[dab.events.DABEvent](#)

|

+--*OperationControlCnfEvent*

public Class *OperationControlCnfEvent*

extends [DABEvent](#)

OperationControlCnfEvent is generated in response to a *operationControlReq* request.

Version:

1.01

See Also:

[operationControlCnf](#)

Constructor Summary		Page
protected	OperationControlCnfEvent (dab.DABSource source, int result, int attribute, java.lang.Object value) Create a <i>OperationControlCnfEvent</i> object.	437

Method Summary		Page
int	getAttribute () Returns the attribute of the receiver that was involved (cf.	437
int	getResult () Returns the status of the <i>OperationControl</i> command.	447
java.lang.Object	getValue () Returns a copy of the attribute's value.	437

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

OperationControlCnfEvent

```
protected OperationControlCnfEvent(dab.DABSource source,  
                                     int result,  
                                     int attribute,  
                                     java.lang.Object value)
```

Create a `OperationControlCnfEvent` object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the `OperationControl` command. If it is equal to `DABConstants.resultOK`, the command was successful. Otherwise an error has occurred.

getAttribute

```
public int getAttribute()
```

Returns the attribute of the receiver that was involved (cf. `DABConstants.operationControl*`)

getValue

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

Returns a copy of the attribute's value. This is either the actual value, when a read request was issued, or the former value when a change request was issued.

- `DABConstants.operationControlSetVolume`: The former volume of the receiver is returned. It is of type `Integer` in the range from 0 to 100 (percent).
- `DABConstants.operationControlGetVolume`: The current volume of the receiver is returned. It is of type `Integer` in the range from 0 to 100 (percent).
- `DABConstants.operationControlSetServiceFollowing`: The former state of the service following feature is returned. It is of type `Boolean`: true indicates that the service following was switched on, false indicates that it was switched off.
- `DABConstants.operationControlGetServiceFollowing`: The current state of the service following feature is returned. It is of type `Boolean`: true indicates that the service following is switched on, false indicates that it is switched off.
- `DABConstants.operationControlGetServiceFollowingNotifications`: null is returned.
- `DABConstants.operationControlSetDRCMODE`: The former state of the `DRCMODE` following feature is returned. It is of type `Boolean`: true indicates that the `DRCMODE` was switched on, false indicates that it was switched off.
- `DABConstants.operationControlGetDRCMODE`: The current state of `DRC` mode is returned. It is of type `Boolean`: true indicates that the `DRC` mode is switched on, false indicates that it is switched off.
- `DABConstants.operationControlGetDRCMODENotifications`: null is returned.

See Also:

[operationControlReq](#)

Class ReceptionInfoNtfEvent

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.ReceptionInfoNtfEvent

```

public Class **ReceptionInfoNtfEvent**

extends [DABEvent](#)

ReceptionInfoNtfEvent is generated in response to a selectReceptionInfoReq request.

Version:

1.01

See Also:

[receptionInfoNtf](#)

Constructor Summary		Page
protected	ReceptionInfoNtfEvent (dab.DABSource source, int updateFlags, int synchronizationState, int bitErrorRateState, int muteState) Creates a ReceptionInfoNtfEvent object.	435

Method Summary		Page
int	getBitErrorRateState () Returns the bit error rate state.	435
int	getMuteState () Returns the mute state.	436
int	getSynchronizationState () Returns the synchronization state.	435
int	getUpdateFlags () Returns the updateFlags.	435

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ReceptionInfoNtfEvent

```
protected ReceptionInfoNtfEvent(dab.DABSource source,  
                                int updateFlags,  
                                int synchronizationState,  
                                int bitErrorRateState,  
                                int muteState)
```

Creates a ReceptionInfoNtfEvent object.

Method Detail

getUpdateFlags

```
public int getUpdateFlags()
```

Returns the updateFlags. This value is a flag field which indicates if synchronization, biterrrorate and/or mute state has changed. The following values are supported:

- DABConstants.syncUpdateSynchronizationState: The synchronization state has changed. The new state is specified by synchronizationState.
- DABConstants.syncUpdateBitErrorRateState: The biterrrorate state has changed. The new state is specified by bitErrorRateState.
- DABConstants.syncUpdateMuteState: The mute state has changed. The new state is specified by muteState.

getSynchronizationState

```
public int getSynchronizationState()
```

Returns the synchronization state. This value specifies the current synchronization state of the DAB Receiver. The following values are supported:

- DABConstants.stateSynchronizationStateUnknown: The synchronization state is not known.
- DABConstants.stateNotSynchronized: The DAB Receiver is not synchronized. This is the lowest level of synchronization.
- DABConstants.stateDABSignalDetected: The DAB Receiver has detected a DAB Signal.
- DABConstants.stateTimeAndFrequencySynchronized: The DAB Receiver is time and frequency synchronized.
- DABConstants.stateFICReadable: The Service Information channel is readable. This is the highest level of synchronization.

getBitErrorRateState

```
public int getBitErrorRateState()
```

Returns the bit error rate state. This value specifies the current biterrrorate state. The following values are supported:

- DABConstants.bitErrorRateLevelUnknown: The current biterrrorate is unknown.
- DABConstants.bitErrorRateLevel1: The biterrrorate is smaller than 5e-4.
- DABConstants.bitErrorRateLevel2: The biterrrorate is smaller than 5e-3.
- DABConstants.bitErrorRateLevel3: The biterrrorate is smaller than 5e-2.
- DABConstants.bitErrorRateLevel4: The biterrrorate is smaller than 1e-1.
- DABConstants.bitErrorRateLevel5: The biterrrorate is equal or larger than 1e-1.

getMuteState

```
public int getMuteState()
```


Returns the mute state. This value specifies the current mute state. The following values are supported:

- DABConstants.muteStateUnknown: The current mute state is unknown.
- DABConstants.muteStateMuting: The DAB Receiver is permanently muting.
- DABConstants.muteStatePartialMuting: Some audio frames were muted.
- DABConstants.muteStateNotMuting: No frame was muted.

Class RespondConflictResolutionCnfEvent

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.RespondConflictResolutionCnfEvent

```

public Class **RespondConflictResolutionCnfEvent**

extends [DABEvent](#)

Constructor Summary		Page
protected	RespondConflictResolutionCnfEvent (dab.DABSource source, int _result)	433

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
int	getResult () Returns the status of the respondConflictResolution command.	447

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

RespondConflictResolutionCnfEvent

```
protected RespondConflictResolutionCnfEvent(dab.DABSource source,
                                             int _result)
```

Method Detail

getResult

```
public int getResult()
```

Returns the status of the respondConflictResolution command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

Class ScanCnfEvent

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.ScanCnfEvent

```

public Class **ScanCnfEvent**

extends [DABEvent](#)

ScanCnfEvent is generated in response to a scanReq request.

Version:

1.01

See Also:

[scanReq](#)

Constructor Summary		Page
protected	ScanCnfEvent (dab.DABSource source, int result, int tuneState, int tuneFrequency, int transmissionModes, int noOfEnsemblesFound) Creates an ScanCnfEvent object.	432

Method Summary		Page
int	getNoOfEnsemblesFound () Returns the number of DAB Ensembles that have been found during the execution of the scan command.	433
int	getResult () Returns the status of the Scan command.	447
int	getTransmissionModes () Returns the transmission modes a DAB Receiver should look for DAB Ensembles.	432
int	getTuneFrequency () Returns the currently tuned frequency.	432
int	getTuneState () Returns the current tune state.	432

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

ScanCnfEvent

```
protected ScanCnfEvent(dab.DABSource source,  
                        int result,  
                        int tuneState,  
                        int tuneFrequency,  
                        int transmissionModes,  
                        int noOfEnsemblesFound)
```

Creates an `ScanCnfEvent` object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the Scan command. If it is equal to `DABConstants.resultOK`, the command was successful. Otherwise an error has occurred.

getTuneState

```
public int getTuneState()
```

Returns the current tune state. The following values are supported:

- `DABConstants.tuneStateNotTuned`: The DAB Receiver is not tuned to a known frequency.
- `DABConstants.tuneStateTuned`: The DAB Receiver is tuned to a frequency specified by `tuneFrequency`.

getTuneFrequency

```
public int getTuneFrequency()
```

Returns the currently tuned frequency.

getTransmissionModes

```
public int getTransmissionModes()
```

Returns the transmission modes a DAB Receiver should look for DAB Ensembles. The default value is `DABConstants.transmissionModeAutomatic` which means that the receiver is automatically detecting the `Transmissionmode`. The returned value is a flag field supporting the following flags which can be specified together:

- `DABConstants.transmissionModeAutomatic`: The `Transmissionmode` is automatically detected
- `DABConstants.transmissionMode1`: At the specified frequency it is tested if a DAB Ensemble is sent in `Transmissionmode 1`.

- DABConstants.transmissionMode2: At the specified frequency it is tested if a DAB Ensemble is sent in Transmissionmode 2.
- DABConstants.transmissionMode3: At the specified frequency it is tested if a DAB Ensemble is sent in Transmissionmode 3.
- DABConstants.transmissionMode4: At the specified frequency it is tested if a DAB Ensemble is sent in Transmissionmode 4.

getNoOfEnsemblesFound

```
public int getNoOfEnsemblesFound()
```

Returns the number of DAB Ensembles that have been found during the execution of the scan command.

Class ScanNtfEvent

[dab.events](#)

```
java.lang.Object
```

```
|
+--dab.events.DABEvent
    |
    +--dab.events.ScanNtfEvent
```

```
public Class ScanNtfEvent
```

```
extends DABEvent
```

ScanNtfEvent is generated in response to a scanReq request.

Version:

1.01

See Also:

[scanReq](#)

Constructor Summary		Page
protected	ScanNtfEvent (dab.DABSource source, int tuneFrequency, int notifications) Creates a ScanNtfEvent object.	430

Method Summary		Page
int	getNotifications () Returns the notification type.	430
int	getTuneFrequency () Returns the currently tuned frequency in Hertz.	432

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

ScanNtfEvent

```
protected ScanNtfEvent(dab.DABSource source,  
                        int tuneFrequency,  
                        int notifications)
```

Creates a `ScanNtfEvent` object.

Method Detail

getTuneFrequency

```
public int getTuneFrequency()
```

Returns the currently tuned frequency in Hertz.

getNotifications

```
public int getNotifications()
```

Returns the notification type. The returned value is a flag field supporting the following flags which can be specified together:

- `DABConstants.notifications16kHzSteps`: A 16 kHz step has been made.
- `DABConstants.notificationsTableEntry`: A frequency of the specified frequency table has been reached.
- `DABConstants.notificationsEnsembleFound`: A DAB Ensemble has been found.

Class **SearchCnfEvent**

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.SearchCnfEvent
```

```
public Class SearchCnfEvent
```

```
extends DABEvent
```

`SearchCnfEvent` is generated in response to a `searchReq` request.

Version:

1.01

See Also:[searchCnf](#)

Constructor Summary		Page
protected	SearchCnfEvent (dab.DABSource source, int result, int tuneState, int tuneFrequency, int transmissionMode, int synchronizationState) Create a SearchCnfEvent	428

Method Summary		Page
int	getResult () Returns the result.	447
int	getSynchronizationState () Returns the synchronization state.	435
int	getTransmissionMode () Returns the transmission mode.	428
int	getTuneFrequency () Return the tune frequency in use.	432
int	getTuneState () Return the tune state.	432

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
<code>equals</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>

Constructor Detail

SearchCnfEvent

```
protected SearchCnfEvent(dab.DABSource source,
    int result,
    int tuneState,
    int tuneFrequency,
    int transmissionMode,
    int synchronizationState)
```

Create a SearchCnfEvent

Method Detail

getResult

```
public int getResult()
```

Returns the result. This value indicates the status of the Tune command. If it is equal to `DABConstants.resultOK`, the command was successful. Otherwise an error has occurred.

getTuneState

```
public int getTuneState()
```

Return the tune state. This value indicates the current tune state independent from the command result indicated by result. The following values are supported:

- `DABConstants.stateNotTuned`: The DAB receiver is not tuned to a known frequency. An error has occurred in this case and the following parameters are undefined.
 - `DABConstants.stateTuned`: The DAB receiver is tuned to a frequency specified by `tuneFrequency` and the following parameters are defined.
-

getTuneFrequency

```
public int getTuneFrequency()
```

Return the tune frequency in use.

getTransmissionMode

```
public int getTransmissionMode()
```

Returns the transmission mode. This value specifies the DAB transmission mode the DAB receiver has detected. The following values are supported:

- `DABConstants.transmissionMode1`: The found DAB Ensemble is sent in Transmissionmode 1.
 - `DABConstants.transmissionMode2`: The found DAB Ensemble is sent in Transmissionmode 2.
 - `DABConstants.transmissionMode3`: The found DAB Ensemble is sent in Transmissionmode 3.
 - `DABConstants.transmissionMode4`: The found DAB Ensemble is sent in Transmissionmode 4.
 - `DABConstants.transmissionModeUnknown`: The transmission mode is unknown.
-

getSynchronizationState

```
public int getSynchronizationState()
```

Returns the synchronization state. This value specifies the current synchronization state of the DAB Receiver. The following values are supported:

- `DABConstants.stateNotSynchronized`: The DAB Receiver is not synchronized. This is the lowest level of synchronization.
- `DABConstants.stateDABSignalDetected`: The DAB Receiver has detected a DAB Signal.
- `DABConstants.stateTimeAndFrequencySynchronized`: The DAB Receiver is time and frequency synchronized.
- `DABConstants.stateFICReadable`: The Service Information channel is readable. This is the highest level of synchronization.

Class SearchNtfEvent

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.SearchNtfEvent

```

public Class **SearchNtfEvent**

extends [DABEvent](#)

SearchNtfEvent is generated in response to a searchReq request.

Version:

1.01

See Also:

[searchNtf](#)

Constructor Summary		Page
protected	SearchNtfEvent (dab.DABSource source, int tuneFrequency, int notifications) Creates a SearchNtfEvent.	426

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
int	getNotifications () Returns the notification type.	430
int	getTuneFrequency () Returns the currently tuned frequency in Hertz.	432

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

SearchNtfEvent

```
protected SearchNtfEvent(dab.DABSource source,
                          int tuneFrequency,
                          int notifications)
```

Creates a SearchNtfEvent.

Method Detail**getTuneFrequency**

```
public int getTuneFrequency()
```

Returns the currently tuned frequency in Hertz.

getNotifications

```
public int getNotifications()
```

Returns the notification type. The value is a flag field supporting the following flags which can be specified together:

- notifications16kHzSteps: A 16 kHz step has been made.
- notificationsTableEntry: A frequency of the specified frequency table has been reached.
- notificationsSearchStarted: Searching for a DAB Ensemble has been started.

Class SelectApplicationCnfEvent

[dab.events](#)

```
java.lang.Object
```

```
|
+--dab.events.DABEvent
    |
    +--dab.events.SelectApplicationCnfEvent
```

```
public Class SelectApplicationCnfEvent
```

```
extends DABEvent
```

SelectApplicationCnfEvent is generated in response to a selectApplicationReq request.

Version:

0.2

See Also:

[selectApplicationReq](#)

Constructor Summary		Page
protected	SelectApplicationCnfEvent (dab.DABSource source, int result, dab.DABAppProxy proxy) Creates an SelectApplicationCnfEvent object.	425

Method Summary		Page
dab.DABAppProxy	getApplicationProxy () Returns the proxy for the loaded application.	425

int	getResult()	447
	Returns the status of the SelectApplication command.	

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

SelectApplicationCnfEvent

```
protected SelectApplicationCnfEvent(dab.DABSource source,
                                     int result,
                                     dab.DABAppProxy proxy)
```

Creates an SelectApplicationCnfEvent object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the SelectApplication command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getApplicationProxy

```
public dab.DABAppProxy getApplicationProxy()
```

Returns the proxy for the loaded application. The value is null, when result != DABConstants.resultOK

Class **SelectComponentCnfEvent**

[dab.events](#)

java.lang.Object

|

+--[dab.events.DABEvent](#)

|

+--**dab.events.SelectComponentCnfEvent**

```
public Class SelectComponentCnfEvent
```

```
extends DABEvent
```

The *SelectComponentCnfEvent* is generated in response to a *selectComponentReq* request.

Version:

1.02

See Also:

[selectComponentCnf](#)

Constructor Summary		Page
protected	SelectComponentCnfEvent (dab.DABSource source, int result, dab.si.ComponentId componentId, int selectionMode) Create a <i>SelectComponentCnfEvent</i> object.	423

Method Summary		Page
dab.si.ComponentId	GetComponentId () Returns the component which delivers the started or stopped application.	452
int	getResult () Returns the status of the <i>SelectComponent</i> command.	447
int	getSelectionMode () Return the selection mode for the component.	452

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SelectComponentCnfEvent

```
protected SelectComponentCnfEvent(dab.DABSource source,
                                   int result,
                                   dab.si.ComponentId componentId,
                                   int selectionMode)
```

Create a *SelectComponentCnfEvent* object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the *SelectComponent* command. If it is equal to *DABConstants.resultOK*, the command was successful. Otherwise an error has occurred.

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the component which delivers the started or stopped application.

getSelectionMode

```
public int getSelectionMode()
```

Return the selection mode for the component. The following flags are supported:

- `DABConstants.selectionModeReplace`: All former selected components of the same type are stopped and the specified component is started. The same type means an audio component replaces any other selected audio component, a data component replaces all other selected independent data components and a programme-associated data component replaces all other selected programme-associated data components.
- `DABConstants.selectionModeAdd`: The application delivered by the specified component is started. Other selected components are not affected.
- `DABConstants.selectionModeRemove`: The selection of the specified component is removed.
- `DABConstants.selectionModeRemoveAll`: All existing component selections are removed. The parameter `serviceId` is set to null in this case.

Class *SelectComponentStreamCnfEvent*

[dab.events](#)

```
java.lang.Object
```

```

|
+--dab.events.DABEvent
    |
    +--dab.events.SelectComponentStreamCnfEvent

```

```
public Class SelectComponentStreamCnfEvent
```

```
extends DABEvent
```

`SelectComponentStreamCnfEvent` is generated in response to a `selectComponentStreamReq`.

Version:

1.02

Constructor Summary		<i>Page</i>
protected	SelectComponentStreamCnfEvent (<code>dab.DABSource source, int result, int streamType, java.io.InputStream stream</code>)	422

Method Summary		<i>Page</i>
int	getResult () Returns the status of the <code>SelectComponentStream</code> command.	447

java.io.InputStream	getStream() Returns the stream.	422
int	getStreamType() Returns the type of the stream (cf.	422

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

SelectComponentStreamCnfEvent

```
protected SelectComponentStreamCnfEvent(dab.DABSource source,  
                                           int result,  
                                           int streamType,  
                                           java.io.InputStream stream)
```

Method Detail

getResult

```
public int getResult()
```

Returns the status of the SelectComponentStream command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getStreamType

```
public int getStreamType()
```

Returns the type of the stream (cf. DABConstants.streamType*).

getStream

```
public java.io.InputStream getStream()
```

Returns the stream.

Class *SelectObjectCnfEvent*

[dab.events](#)

java.lang.Object

|

+--[dab.events.DABEvent](#)

|

+--**dab.events.SelectObjectCnfEvent**

public Class **SelectObjectCnfEvent**

extends [DABEvent](#)

The *SelectObjectCnfEvent* is generated in response to a *selectObjectReq* request.

Version:

1.02

See Also:

[selectObjectReq](#)

Constructor Summary		Page
protected	SelectObjectCnfEvent (dab.DABSource source, int result, dab.si.ComponentId componentId, dab.data.ObjectId objectId, int requestMode, boolean replaceSelections, java.util.Date accessTime) Creates a <i>SelectObjectCnfEvent</i> object.	420

Method Summary		Page
java.util.Date	getAccessTime () Returns the expected relative access time for delivery of the object.	421
dab.si.ComponentId	getComponentId () Returns the component the object is belonging to.	452
dab.data.ObjectId	getObjectId () Returns the selected object	439
boolean	getReplaceSelections () Returns all current object selections belonging to the component identified by serviceInfoId are replaced with this selection.	421
int	getRequestMode () Returns the current selection mode for the specified object.	420
int	getResult () Returns the status of the <i>SelectObject</i> command.	447

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

SelectObjectCnfEvent

```
protected SelectObjectCnfEvent(dab.DABSource source,
                                int result,
                                dab.si.ComponentId componentId,
                                dab.data.ObjectId objectId,
                                int requestMode,
                                boolean replaceSelections,
                                java.util.Date accessTime)
```

Creates a `SelectObjectCnfEvent` object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the `SelectObject` command. If it is equal to `DABConstants.resultOK`, the command was successful. Otherwise an error has occurred.

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the component the object is belonging to.

getObjectId

```
public dab.data.ObjectId getObjectId()
```

Returns the selected object

getRequestMode

```
public int getRequestMode()
```

Returns the current selection mode for the specified object.

- `DABConstants.requestModeOff`: The object selection is removed.
- `DABConstants.requestModeOnce`: The object is requested for one-time delivery. After the first reception from the broadcast channel the object is delivered to the connected DAB client. The client is not notified about new versions.
- `DABConstants.requestModeUpdate`: The object is requested for update delivery. After the first reception from the broadcast channel the object is delivered to the connected client. Additionally each new version of the object is delivered.

getReplaceSelections

```
public boolean getReplaceSelections()
```

Returns all current object selections belonging to the component identified by serviceInfoId are replaced with this selection. If the returned value is true, then all selections are removed. If the returned value is false, then existing selections remain unchanged.

getAccessTime

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

Returns the expected relative access time for delivery of the object.

Class *SelectReceptionInfoCnfEvent*

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.SelectReceptionInfoCnfEvent
```

```
public Class SelectReceptionInfoCnfEvent
```

```
extends DABEvent
```

The *SelectReceptionInfoCnfEvent* is generated in response to a *selectReceptionInfoReq* request.

Version:

1.01

See Also:

[selectReceptionInfoReq](#)

Constructor Summary		Page
protected	SelectReceptionInfoCnfEvent (dab.DABSource source, int result, boolean synchronizationNotifications, boolean bitErrorRateNotifications, boolean muteStateNotifications) Creates a <i>SelectReceptionInfoCnfEvent</i> object.	418

Method Summary		Page
boolean	getBitErrorRateNotifications () Returns bit error rate notifications.	418

boolean	<u>getMuteStateNotifications()</u> Returns mute state notifications.	418
int	<u>getResult()</u> Returns the status of the SelectReceptionInfo command.	447
boolean	<u>getSynchronizationNotifications()</u> Returns synchronization notifications.	418

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

SelectReceptionInfoCnfEvent

```
protected SelectReceptionInfoCnfEvent(dab.DABSource source,  
                                       int result,  
                                       boolean synchronizationNotifications,  
                                       boolean bitErrorRateNotifications,  
                                       boolean muteStateNotifications)
```

Creates a SelectReceptionInfoCnfEvent object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the SelectReceptionInfo command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getSynchronizationNotifications

```
public boolean getSynchronizationNotifications()
```

Returns synchronization notifications. The value specifies if the client is notified about state changes concerning DAB signal synchronization. If the returned value is true notifications are provided, if it is false no notifications are provided.

getBitErrorRateNotifications

```
public boolean getBitErrorRateNotifications()
```

Returns bit error rate notifications. The value specifies if the client is notified about state changes concerning the biterrorate. If the returned is true notifications are provided, if it is false no notifications are provided.

getMuteStateNotifications

```
public boolean getMuteStateNotifications()
```

Returns mute state notifications. This value specifies if the client is notified about state changes concerning the mute state of the audio decoder. If the returned value is true notifications are provided, if it is false no notifications are provided.

Class **SelectSICnfEvent**

dab.events

```
java.lang.Object
```

```

|
+--dab.events.DABEvent
|
+--dab.events.SelectSICnfEvent

```

```
public Class SelectSICnfEvent
```

```
extends DABEvent
```

The SelectSICnfEvent is generated in response to a selectSIReq request.

Version:

1.02

See Also:

[selectSICnf](#)

Constructor Summary		<i>Page</i>
protected	SelectSICnfEvent (dab.DABSource source, int result, boolean ensembleInfo, boolean serviceInfo, boolean componentInfo, boolean autoDelivery) Creates an SelectSICnfEvent.	415

Method Summary		<i>Page</i>
boolean	getAutoDelivery () Returns auto delivery.	416
boolean	getComponentInfo () Returns component info.	447
boolean	getEnsembleInfo () Returns the ensemble info.	446
int	getResult () Returns the status of the SelectSI command.	447

boolean	getServiceInfo() Returns service info.	442
---------	---	-----

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

SelectSICnfEvent

```
protected SelectSICnfEvent(dab.DABSource source,
                           int result,
                           boolean ensembleInfo,
                           boolean serviceInfo,
                           boolean componentInfo,
                           boolean autoDelivery)
```

Creates an SelectSICnfEvent.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the SelectSI command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getEnsembleInfo

```
public boolean getEnsembleInfo()
```

Returns the ensemble info. This value specifies if the DAB client is subscribed to ensemble-specific notifications. The following values are supported:

- true: The client is notified about ensembleAdded, ensembleChanged and ensembleRemoved events.
- false: The client is not notified about ensembleAdded, ensembleChanged and ensembleRemoved events.

getServiceInfo

```
public boolean getServiceInfo()
```

Returns service info. This value specifies if the DAB client is subscribed to service-specific notifications. The following values are supported:

- true: The client is notified about serviceAdded, serviceChanged and serviceRemoved events.
- false: The client is not notified about serviceAdded, serviceChanged and serviceRemoved events.

getComponentInfo

```
public boolean getComponentInfo()
```

Returns component info. This value specifies if the client is subscribed to component-specific notifications. The following values are supported:

- true: The client is notified about componentAdded, componentChanged and componentRemoved events.
- false: The client is not notified about componentAdded, componentChanged and componentRemoved events.

getAutoDelivery

```
public boolean getAutoDelivery()
```

Returns auto delivery. This value specifies if the information related to the notification is sent together with the notification (serviceInfoNtf) or not. The following values are supported:

- true: The serviceInfoNtf method delivers the notification together with the information object. This is only possible for -Added and -Changed notifications but not for -Removed because in the latter case the service element is no longer existing.
- false: The serviceInfoNtf method delivers only the notification. The information object (EnsembleInfo, ServiceInfo or ComponentInfo) itself can be obtained by use of getEnsembleInfoReq, getServiceInfoReq or getComponentInfoReq.

Class ServiceFollowingNtfEvent

[dab.events](#)

```
java.lang.Object
```

```

|
+---dab.events.DABEvent
|
+---dab.events.ServiceFollowingNtfEvent

```

```
public Class ServiceFollowingNtfEvent
```

```
extends DABEvent
```

The ServiceFollowingNtfEvent is generated when a service following action is taken by the receiver.

Version:

1.02

See Also:

[serviceFollowingNtf](#)

Constructor Summary		<i>Page</i>
protected	<pre>ServiceFollowingNtfEvent(dab.DABSource source, int action, dab.si.EnsembleId ensembleId, dab.si.ComponentId componentId)</pre> <p>Creates an ServiceFollowingNtfEvent event.</p>	414

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
int	getAction() Returns the service following action code.	414
dab.si.ComponentId	getComponentId() Returns the service identifier of the component that is involved in the action	452
dab.si.EnsembleId	getEnsembleId() Returns the service identifier of the ensemble that is involved in the action	414

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ServiceFollowingNtfEvent

```
protected ServiceFollowingNtfEvent(dab.DABSource source,
                                   int action,
                                   dab.si.EnsembleId ensembleId,
                                   dab.si.ComponentId componentId)
```

Creates an ServiceFollowingNtfEvent event.

Method Detail

getAction

```
public int getAction()
```

Returns the service following action code. The following codes are possible:

- DABConstants.ServiceFollowingLeavingService: The service following has started. The current ensemble and service are left.
- DABConstants.ServiceFollowingTryingAlternativeService: An alternative frequency is tried.
- DABConstants.ServiceFollowingSelectingService: The service following is finished. The receiver is tuned to a new ensemble and audio service.

getEnsembleId

```
public dab.si.EnsembleId getEnsembleId()
```

Returns the service identifier of the ensemble that is involved in the action

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the service identifier of the component that is involved in the action

Class **SINtfEvent**

[dab.events](#)

```
java.lang.Object
```

```
|
+--dab.events.DABEvent
|
+--dab.events.SINtfEvent
```

```
public Class SINtfEvent
```

```
extends DABEvent
```

The SINtfEvent is generated in response to a selectSIReq request.

Version:

1.04

See Also:

[siNtf](#)

Constructor Summary		Page
protected	SINtfEvent (dab.DABSource source, int notification, int updateFlags, dab.si.SIId serviceInfoId, dab.si.EnsembleInfo ensembleInfo, dab.si.ServiceInfo serviceInfo, dab.si.ComponentInfo componentInfo)	410
	Creates a SINtfEvent object.	

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
dab.si.ComponentInfo	getComponentInfo () <p>If notification signals a component-related notification of type DABConstants.componentAdded or DABConstants.componentChanged and AutoDelivery has been activated with the subscription, then the value refers to a component information object.</p>	447
dab.si.EnsembleInfo	getEnsembleInfo () <p>If notification signals an ensemble-related notification of type DABConstants.ensembleAdded or DABConstants.ensembleChanged and AutoDelivery has been activated with the subscription, then the returned value refers to an ensemble information object.</p>	446
int	getNotification () <p>Returns the notification type.</p>	411

dab.si.ServiceInfo	getServiceInfo() If notification signals a service-related notification of type DABConstants.serviceAdded or DABConstants.serviceChanged and AutoDelivery has been activated with the subscription, then the returned value refers to a service information object.	442
dab.si.SIID	getServiceInfoId() Returns the instance of the service element (Ensemble, Service, Component) that has changed.	412
int	getUpdateFlags() Returns more detailed information about which part of the Service Directory has changed.	435

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SINtfEvent

```
protected SINtfEvent(dab.DABSource source,
                    int notification,
                    int updateFlags,
                    dab.si.SIID serviceInfoId,
                    dab.si.EnsembleInfo ensembleInfo,
                    dab.si.ServiceInfo serviceInfo,
                    dab.si.ComponentInfo componentInfo)
```

Creates a SINtfEvent object.

Method Detail

getNotification

```
public int getNotification()
```

Returns the notification type. The following values are supported:

Ensemble-related notification:

- ensembleAdded: A new DAB Ensemble is available.
- ensembleRemoved: A known DAB Ensemble is no longer available. All dependent services and components are also no longer available.
- ensembleChanged: A known DAB Ensemble has changed which means its attributes have changed. This case signals changes to the ensemble itself and not changes in linking to child services. This means if a child service is added or removed this is not indicated by a ensembleChanged notification.

Service-related notification:

- serviceAdded: A new DAB Service is available.
- serviceRemoved: A known DAB Service is no longer available. All dependent components are also no longer available.
- serviceChanged: A known DAB Service has changed which means its attributes have changed. This case signals changes to the service itself and not changes in linking to child components. This means if a child component is added or removed this is not indicated by a DABConstants.serviceChanged notification.

Component-related notification:

- componentAdded: A new DAB Component is available.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

- `componentRemoved`: A known DAB Component is no longer available.
- `componentChanged`: A known DAB Component has changed which means its attributes have changed.

getUpdateFlags

```
public int getUpdateFlags()
```

Returns more detailed information about which part of the Service Directory has changed. The value is a flag field and supports the following flags depending on the service element type.

In case of a `DABConstants.ensembleAdded` or `DABConstants.ensembleChanged` notification the following values are defined:

- `DABConstants.updatedLabel`: The Ensemble label has changed.
- `DABConstants.updatedCountry`: The Country information which specifies which area is covered by the Ensemble has changed.

In case of a `DABConstants.serviceAdded` or `DABConstants.serviceChanged` notification the following values are defined:

- `DABConstants.updatedLabel`: The Service label has changed.
- `DABConstants.updatedCountry`: The Country information which specifies which area is covered by the Service has changed.
- `DABConstants.updatedTimeOffset`: The time offset for the specified Service has changed.
- `DABConstants.updatedRegion`: The region has changed.
- `DABConstants.updatedStaticProgrammeType`: The static programme type information of the specified audio service has changed.
- `DABConstants.updatedDynamicProgrammeType`: The static programme type information of the specified audio service has changed.
- `DABConstants.updatedAnnouncement`: The announcement information of the specified audio service has changed.
- `DABConstants.updatedLanguage`: The language information of the specified audio service has changed.
- `DABConstants.updatedRegionId`: The region identifier has changed.
- `DABConstants.updatedRegionLabel`: The region label has changed.
- `DABConstants.updatedAnnouncementSupport`: The announcement support information of the specified audio service has changed.
- `DABConstants.updatedProgrammeNumber`: The programme number has changed.

In case of a `DABConstants.componentAdded` or `DABConstants.componentChanged` notification the following values are defined:

- `DABConstants.updatedLabel`: The component label has changed.
- `DABConstants.updatedLanguage`: The language information of the specified audio component has changed.
- `DABConstants.updatedStartObject`: In case of a `BroadcastWebSite` application carried in the related component this indicates that the start object (homepage) is known.
- `DABConstants.updatedObjectDirectory`: The MOT object directory has changed.
- `DABConstants.updatedAudioComponent`: The link to the audio component has changed.
- `DABConstants.updatedBitrate`: The bitrate has changed.

getServiceInfoId

```
public dab.si.SIId getServiceInfoId()
```

Returns the instance of the service element (Ensemble, Service, Component) that has changed. It can be used in order to request the related information object with the `getEnsembleInfo`, `getServiceInfo` or `getComponentInfo` command.

getEnsembleInfo

```
public dab.si.EnsembleInfo getEnsembleInfo()
```

If notification signals an ensemble-related notification of type `DABConstants.ensembleAdded` or `DABConstants.ensembleChanged` and `AutoDelivery` has been activated with the subscription, then the returned value refers to an ensemble information object. If `AutoDelivery` is not activated or this is a service-related or component-related notification then null is returned.

getServiceInfo

```
public dab.si.ServiceInfo getServiceInfo()
```

If notification signals a service-related notification of type `DABConstants.serviceAdded` or `DABConstants.serviceChanged` and `AutoDelivery` has been activated with the subscription, then the returned value refers to a service information object. If `AutoDelivery` is not activated or this is an ensemble-related or component-related notification then null is returned.

getComponentInfo

```
public dab.si.ComponentInfo getComponentInfo()
```

If notification signals a component-related notification of type `DABConstants.componentAdded` or `DABConstants.componentChanged` and `AutoDelivery` has been activated with the subscription, then the value refers to a component information object. If `AutoDelivery` is not activated or this is an ensemble-related or service-related notification then null is returned.

Class SystemFailureNtfEvent[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.SystemFailureNtfEvent
```

```
public Class SystemFailureNtfEvent
```

```
extends DABEvent
```

The System-Failure event is generated when a fatal error has happened in the system.

Version:

1.02

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Summary		Page
protected	SystemFailureNtfEvent (dab.DABSource source, int reason)	409

Method Summary		Page
int	getReason ()	452
Returns the reason for the system failure (the codes are compatible with DABConstants.result*)		

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

SystemFailureNtfEvent

```
protected SystemFailureNtfEvent(dab.DABSource source,
                                int reason)
```

Method Detail

getReason

```
public int getReason()
```

Returns the reason for the system failure (the codes are compatible with DABConstants.result*)

Class TuneCnfEvent

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.TuneCnfEvent
```

```
public Class TuneCnfEvent
```

```
extends DABEvent
```

The TuneCnfEvent is generated in response to a tuneReq request.

Version:

1.01

See Also:

[tuneCnf](#)

Constructor Summary		Page
protected	TuneCnfEvent (dab.DABSource source, int result, int tuneState, int tuneFrequency, int transmissionMode, int synchronizationState) Creates a TuneCnfEvent.	407

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
int	getResult () Returns the status of the Tune command.	447
int	getSynchronizationState () Returns the current synchronization state of the DAB Receiver.	435
int	getTransmissionMode () Returns the DAB transmission mode the DAB receiver has detected.	428
int	getTuneFrequency () Return the frequency currently in use.	432
int	getTuneState () Returns the current tune state independent from the command result indicated by result.	432

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

TuneCnfEvent

```
protected TuneCnfEvent(dab.DABSource source,
                       int result,
                       int tuneState,
                       int tuneFrequency,
                       int transmissionMode,
                       int synchronizationState)
```

Creates a TuneCnfEvent.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the Tune command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getTuneState

```
public int getTuneState()
```

Returns the current tune state independent from the command result indicated by result. The following values are supported:

- `tuneStateNotTuned`: The DAB receiver is not tuned to a known frequency. An error has occurred in this case and the following parameters are undefined.
- `tuneStateTuned`: The DAB receiver is tuned to a frequency specified by `tuneFrequency` and the following parameters are defined.

getTuneFrequency

```
public int getTuneFrequency()
```

Return the frequency currently in use.

getTransmissionMode

```
public int getTransmissionMode()
```

Returns the DAB transmission mode the DAB receiver has detected. The following values are supported:

- `DABConstants.transmissionMode1`: The found DAB Ensemble is sent in Transmissionmode 1.
- `DABConstants.transmissionMode2`: The found DAB Ensemble is sent in Transmissionmode 2.
- `DABConstants.transmissionMode3`: The found DAB Ensemble is sent in Transmissionmode 3.
- `DABConstants.transmissionMode4`: The found DAB Ensemble is sent in Transmissionmode 4.
- `DABConstants.transmissionModeUnknown`: The transmission mode is unknown.

getSynchronizationState

```
public int getSynchronizationState()
```

Returns the current synchronization state of the DAB Receiver. The following values are supported:

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

- `DABConstants.stateNotSynchronized`: The DAB Receiver is not synchronized. This is the lowest level of synchronization.
- `DABConstants.stateDABSignalDetected`: The DAB Receiver has detected a DAB Signal.
- `DABConstants.stateTimeAndFrequencySynchronized`: The DAB Receiver is time and frequency synchronized.
- `DABConstants.stateFICReadable`: The Service Information channel is readable. This is the highest level of synchronization.

Package **dab.si**

Class Summary		<i>Page</i>
ComponentId	The ComponentId is an identifier for a DAB component.	405
ComponentInfo	ComponentInfo is used to represent components.	401
DABURL		387

EnsembleId	The EnsembleId is an identifier for a DAB ensemble.	400
EnsembleInfo	EnsembleInfo represents information about a particular ensemble.	398
ServiceId	The ServiceId is an identifier for a DAB service.	396
ServiceInfo	ServiceInfo is used to represent a service.	391
SIId	The SIId is an identifier for a DAB Ensemble, a DAB Service or a DAB Service Component.	390

Exception Summary		Page
DABInvalidURLException	The DABInvalidURLException class is used to indicate a invalid DAB URL.	386

Class ComponentId

[dab.si](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

java.lang.Object

|

+--[dab.si.SIId](#)

|

+--**dab.si.ComponentId**

public Class **ComponentId**

extends [SIId](#)

The ComponentId is an identifier for a DAB component.

Version:

1.03

Constructor Summary		Page
ComponentId (dab.si.ComponentId id)	Constructs a copy of the given ComponentId object.	406
ComponentId (java.lang.String Id)	Constructs a ComponentId object from the given string.	406

Methods inherited from class dab.si.[SIId](#)

[compareTo](#), [compareTo](#), [getId](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

ComponentId

```
public ComponentId(java.lang.String Id)
```

Constructs a ComponentId object from the given string.

See Also:

[getId](#)

ComponentId

```
public ComponentId(dab.si.ComponentId id)
```

Constructs a copy of the given ComponentId object.

Class ComponentInfo

[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+--dab.si.ComponentInfo
```

```
public Class ComponentInfo
```

```
extends java.lang.Object
```

ComponentInfo is used to represent components.

Version:

1.06

Constructor Summary		Page
protected	ComponentInfo (dab.si.ComponentId id, int type, byte[] data, boolean isPrimary, dab.si.ServiceId[] parentIds, int accessControlSystem, boolean hasLabel, dab.data.Label label, boolean hasLanguage, int language, boolean hasStartObjectId, dab.data.ObjectId startObjectId, boolean hasObjectDirectoryId, dab.data.ObjectId objectDirectoryId, boolean hasAudioComponent, dab.si.ComponentId audioComponent, boolean hasBitrate, int bitrate)	403

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
int	getAccessControlSystem ()	403
	Returns the access control system of the component (cf.	

dab.si.ComponentId	getAudioComponent () Returns the SIIId of the related audio component.	405
int	getBitrate () Returns the maximum bitrate of the component in bits per second.	404
byte[]	getData () Returns the application specific data of the component (i.e. the user application data).	403
dab.si.ComponentId	getId () Returns the id of the component	403
dab.data.Label	getLabel () Returns the label of the component	404
int	getLanguage () Returns the language information (cf.	404
dab.data.ObjectId	getObjectDirectoryId () Returns the id of the object directory	404
dab.si.ServiceId[]	getParentIds () Returns a reference to the ids of the parents	404
dab.data.ObjectId	getStartObjectId () Returns the id of the start object	404
int	getType () Returns the type of the component.	403
boolean	isPrimary () Indicates whether the component is primary or not	403

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

ComponentInfo

```
protected ComponentInfo(dab.si.ComponentId id,
                          int type,
                          byte[] data,
                          boolean isPrimary,
                          dab.si.ServiceId[] parentIds,
                          int accessControlSystem,
                          boolean hasLabel,
                          dab.data.Label label,
                          boolean hasLanguage,
                          int language,
                          boolean hasStartObjectId,
                          dab.data.ObjectId startObjectId,
                          boolean hasObjectDirectoryId,
                          dab.data.ObjectId objectDirectoryId,
                          boolean hasAudioComponent,
                          dab.si.ComponentId audioComponent,
                          boolean hasBitrate,
                          int bitrate)
```

Method Detail

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

getId

```
public dab.si.ComponentId getId()
```

Returns the id of the component

getType

```
public int getType()
```

Returns the type of the component. This is essentially the user application type (cf. DABConstants.componentType*)

getData

```
public byte[] getData()
```

Returns the application specific data of the component (i.e. the user application data).

isPrimary

```
public boolean isPrimary()
```

Indicates whether the component is primary or not

getAccessControlSystem

```
public int getAccessControlSystem()
```


Returns the access control system of the component (cf. DABConstants.acs*)

getParentIds

```
public dab.si.ServiceId[] getParentIds()
```

Returns a reference to the ids of the parents

getLabel

```
public dab.data.Label getLabel()  
    throws DABNotAvailableException
```

Returns the label of the component

Throws:

[DABNotAvailableException](#) - when the label is not available

getLanguage

```
public int getLanguage()  
    throws DABNotAvailableException
```

Returns the language information (cf. DABConstants.language*)

Throws:

[DABNotAvailableException](#) - when the information is not available

getStartObjectId

```
public dab.data.ObjectId getStartObjectId()  
    throws DABNotAvailableException
```

Returns the id of the start object

Throws:

[DABNotAvailableException](#) - when the start object is not available

getObjectDirectoryId

```
public dab.data.ObjectId getObjectDirectoryId()  
    throws DABNotAvailableException
```

Returns the id of the object directory

Throws:

[DABNotAvailableException](#) - when the object directory is not available

getBitrate

```
public int getBitrate()  
    throws DABNotAvailableException
```

Returns the maximum bitrate of the component in bits per second.

Throws:

[DABNotAvailableException](#) - when the bitrate is not available

getAudioComponent

```
public dab.si.ComponentId getAudioComponent()
    throws DABNotAvailableException
```

Returns the SIIId of the related audio component. Note, that the object has to be a PAD component; otherwise null is returned.

Throws:

[DABNotAvailableException](#) - when not available

Class DABInvalidURLException

[dab.si](#)

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--dab.DABException
            |
            +--dab.si.DABInvalidURLException
```

All Implemented Interfaces:

java.io.Serializable

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

public Class **DABInvalidURLException**

extends [DABException](#)

The DABInvalidURLException class is used to indicate a invalid DAB URL.

Version:

0.01

Constructor Summary	Page
DABInvalidURLException ()	386

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail**DABInvalidURLExceptionion**public **DABInvalidURLExceptionion**()**Class DABURL**[dab.si](#)

java.lang.Object

|

+--**dab.si.DABURL**public Class **DABURL**

extends java.lang.Object

Constructor Summary	Page
DABURL (dab.si.ComponentId id) Constructs a DABURL object from the given component <i>id</i> .	388
DABURL (dab.si.EnsembleId id) Constructs a DABURL object from the given ensemble <i>id</i> .	388
DABURL (dab.si.ServiceId id) Constructs a DABURL object from the given service <i>id</i> .	388
DABURL (java.lang.String daburl) Constructs a DABURL object from the given string.	388

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary	Page
dab.si.ComponentId getComponentId () returns the component id of the DAB URL	452
dab.si.DABURL getComponentURL () returns the DAB URL of a component	389
dab.si.EnsembleId getEnsembleId () returns the ensemble id of the DAB URL	414

dab.si.DABURL	getEnsembleURL () returns the DAB URL representing the ensemble	388
dab.si.ServiceId	getServiceId () returns the service id of the DAB URL	389
dab.si.DABURL	getServiceURL () returns the service id of the DAB URL	389
java.lang.String	toString () returns DAB URL as string	389

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

DABURL

```
public DABURL(java.lang.String daburl)
    throws DABInvalidURLException
```

Constructs a DABURL object from the given string.

Throws:

[DABInvalidURLException](#) - if *daburl* represents an invalid DAB URL

DABURL

```
public DABURL(dab.si.EnsembleId id)
    throws DABInvalidURLException
```

Constructs a DABURL object from the given ensemble *id*.

Throws:

[DABInvalidURLException](#) - if *id* represents an invalid ensemble id

DABURL

```
public DABURL(dab.si.ServiceId id)
    throws DABInvalidURLException
```

Constructs a DABURL object from the given service *id*.

Throws:

[DABInvalidURLException](#) - if *id* represents an invalid service id

DABURL

```
public DABURL(dab.si.ComponentId id)
    throws DABInvalidURLException
```

Constructs a DABURL object from the given component *id*.

Throws:

[DABInvalidURLException](#) - if *id* represents an invalid component id

Method Detail

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

getEnsembleURL

```
public dab.si.DABURL getEnsembleURL()  
    throws DABNotAvailableException
```

returns the DAB URL representing the ensemble

Throws:

[DABNotAvailableException](#) - no ensemble in DAB URL available

getEnsembleId

```
public dab.si.EnsembleId getEnsembleId()  
    throws DABNotAvailableException
```

returns the ensemble id of the DAB URL

Throws:

[DABNotAvailableException](#) - if no ensemble in DAB URL available

getServiceURL

```
public dab.si.DABURL getServiceURL()  
    throws DABNotAvailableException
```

returns the service id of the DAB URL

Throws:

[DABNotAvailableException](#) - if no service in DAB URL available

getServiceId

```
public dab.si.ServiceId getServiceId()  
    throws DABNotAvailableException
```

returns the service id of the DAB URL

Throws:

[DABNotAvailableException](#) - if no service in DAB URL available

getComponentURL

```
public dab.si.DABURL getComponentURL()  
    throws DABNotAvailableException
```

returns the DAB URL of a component

Throws:

[DABNotAvailableException](#) - if no component in DAB URL available

getComponentId

```
public dab.si.ComponentId getComponentId()  
    throws DABNotAvailableException
```

returns the component id of the DAB URL

Throws:

[DABNotAvailableException](#) - no component in DAB URL available

toString

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

returns DAB URL as string

Overrides:

toString in class java.lang.Object

Class EnsembleId

[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+--dab.si.SIID
```

```
|
```

```
+--dab.si.EnsembleId
```

```
public Class EnsembleId
```

```
extends SIID
```

The EnsembleId is an identifier for a DAB ensemble.

Version:

1.03

Constructor Summary	Page
EnsembleId (dab.si.EnsembleId id) Constructs a copy of the given EnsembleId object.	401
EnsembleId (java.lang.String Id) Constructs a EnsembleId object from the given string.	401

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class [dab.si.SIID](#)

```
compareTo, compareTo, getId
```

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail**EnsembleId**

```
public EnsembleId(java.lang.String Id)
```

Constructs a EnsembleId object from the given string.

See Also:

[getId](#)

EnsembleId

```
public EnsembleId(dab.si.EnsembleId id)
```

Constructs a copy of the given EnsembleId object.

Class EnsembleInfo

[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+--dab.si.EnsembleInfo
```

```
public Class EnsembleInfo
```

```
extends java.lang.Object
```

EnsembleInfo represents information about a particular ensemble.

Version:

1.03

Constructor Summary		Page
protected	EnsembleInfo (dab.si.EnsembleId id, dab.si.ServiceId[] serviceIds, int frequency, int transmissionMode, boolean hasDate, java.util.Date date, boolean hasLabel, dab.data.Label label, boolean hasCountry, int country)	399

Method Summary		Page
int	getCountry () Returns country information about the ensemble (cf.	400
java.util.Date	getDate () Returns date and time associated with the ensemble (given as local time)	400

int	getFrequency() Returns the frequency of the ensemble in Hz	399
dab.si.EnsembleId	getId() Returns the id of the ensemble	403
dab.data.Label	getLabel() Returns the label of the ensemble	404
dab.si.ServiceId[]	getServiceIds() Returns a reference to the ids for the services that are contained in the ensemble	399
int	getTransmissionMode() Returns the transmission Mode (cf.	428

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

EnsembleInfo

```
protected EnsembleInfo(dab.si.EnsembleId id,
    dab.si.ServiceId[] serviceIds,
    int frequency,
    int transmissionMode,
    boolean hasDate,
    java.util.Date date,
    boolean hasLabel,
    dab.data.Label label,
    boolean hasCountry,
    int country)
```

Method Detail

getServiceIds

```
public dab.si.ServiceId[] getServiceIds()
```

Returns a reference to the ids for the services that are contained in the ensemble

getId

```
public dab.si.EnsembleId getId()
```

Returns the id of the ensemble

getFrequency

```
public int getFrequency()
```


Returns the frequency of the ensemble in Hz

getTransmissionMode

```
public int getTransmissionMode()
```

Returns the transmission Mode (cf. DABConstants.transmissionMode*)

getLabel

```
public dab.data.Label getLabel()  
    throws DABNotAvailableException
```

Returns the label of the ensemble

Throws:

[DABNotAvailableException](#) - if the label is not available

getCountry

```
public int getCountry()  
    throws DABNotAvailableException
```

Returns country information about the ensemble (cf. DABConstants.country*)

Throws:

[DABNotAvailableException](#) - if the country information is not available

getDate

```
public java.util.Date getDate()  
    throws DABNotAvailableException
```

Returns date and time associated with the ensemble (given as local time)

Throws:

[DABNotAvailableException](#) - if the date is not available

Class ServiceId

[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+--dab.si.SIId
```

```
|
```

```
+--dab.si.ServiceId
```

```
public Class ServiceId
```

```
extends SIId
```

The ServiceId is an identifier for a DAB service.

Version:

1.05

Constructor Summary		Page
ServiceId (dab.si.ServiceId id)	Constructs a copy of the given ServiceId object.	397
ServiceId (java.lang.String Id)	Constructs a ServiceId object from the given string.	397

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
boolean sameService (dab.si.ServiceId id)	returns true, if id and the called object belong to the same service; otherwise false is returned.	397

Methods inherited from class dab.si. SIId
compareTo , compareTo , getId

Methods inherited from class java.lang.Object
equals , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructor Detail

ServiceId

```
public ServiceId(java.lang.String Id)
```

Constructs a ServiceId object from the given string.

See Also:

[getId](#)

ServiceId

```
public ServiceId(dab.si.ServiceId id)
```

Constructs a copy of the given ServiceId object.

See Also:

[getId](#)

Method Detail

sameService

```
public boolean sameService(dab.si.ServiceId id)
```

returns true, if id and the called object belong to the same service; otherwise false is returned.

Class **ServiceInfo**

[dab.si](#)

java.lang.Object

|

+--**dab.si.ServiceInfo**

public Class **ServiceInfo**

extends java.lang.Object

ServiceInfo is used to represent a service.

Version:

1.04

Constructor Summary		Page
protected	ServiceInfo (dab.si.ServiceId id, int type, dab.si.EnsembleId parent, dab.si.ComponentId[] componentIds, boolean isLocal, int accessControlSystem, boolean hasLabel, dab.data.Label label, boolean hasLanguage, int language, boolean hasIsPrimary, boolean isPrimary, boolean hasRegionId, int regionId, boolean hasRegionLabel, dab.data.Label regionLabel, boolean hasStaticProgrammeType, dab.data.ProgrammeType staticProgrammeType, boolean hasDynamicProgrammeType, dab.data.ProgrammeType dynamicProgrammeType, boolean hasProgrammeNumber, dab.data.ProgrammeNumber programmeNumber, boolean hasTimeOffset, int timeOffset, boolean hasAnnouncementSupport, dab.data.AnnouncementSupport announcementSupport, boolean hasCountry, int country)	393

Method Summary		Page
int	getAccessControlSystem () Returns the access control system (cf.	403
dab.data.AnnouncementSupport	getAnnouncementSupport () Returns the information about announcement support	396
dab.si.ComponentId[]	getComponentIds () Returns a reference to ids of the components of the service	394
int	getCountry () Returns the country information of the service (cf.	400
dab.data.ProgrammeType	getDynamicProgrammeType () Returns the dynamic programme type	395
dab.si.ServiceId	getId () Returns the id of the service	403

dab.data.Label	getLabel () Returns the label of the service	404
int	getLanguage () Returns the language of the service (cf.	404
dab.si.EnsembleId	getParent () Returns the parent ensemble	394
dab.data.ProgrammeNumber	getProgrammeNumber () Returns the programme number	395
int	getRegionId () Returns the region id of the service	395
dab.data.Label	getRegionLabel () Returns the region label of the service	395
dab.data.ProgrammeType	getStaticProgrammeType () Returns the static programme type	395
int	getTimeOffset () Returns the time offset of the service (with respect to the time of the ensemble).	396
int	getType () Returns the service type (cf.	403
boolean	isLocalService () Indicates whether the service is local or not	394
boolean	isPrimaryComponentLanguage () Indicates, whether the language of the service is the language of the primary component	395

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ServiceInfo

```
protected ServiceInfo(dab.si.ServiceId id,
                       int type,
                       dab.si.EnsembleId parent,
                       dab.si.ComponentId[] componentIds,
                       boolean isLocal,
                       int accessControlSystem,
                       boolean hasLabel,
                       dab.data.Label label,
                       boolean hasLanguage,
                       int language,
                       boolean hasIsPrimary,
                       boolean isPrimary,
                       boolean hasRegionId,
                       int regionId,
                       boolean hasRegionLabel,
                       dab.data.Label regionLabel,
                       boolean hasStaticProgrammeType,
                       dab.data.ProgrammeType staticProgrammeType,
                       boolean hasDynamicProgrammeType,
                       dab.data.ProgrammeType dynamicProgrammeType,
                       boolean hasProgrammeNumber,
                       dab.data.ProgrammeNumber programmeNumber,
                       boolean hasTimeOffset,
                       int timeOffset,
                       boolean hasAnnouncementSupport,
                       dab.data.AnnouncementSupport announcementSupport,
                       boolean hasCountry,
                       int country)
```

Method Detail

getId

```
public dab.si.ServiceId getId()
```

Returns the id of the service

getType

```
public int getType()
```

Returns the service type (cf. DABConstants.serviceType*)

getParent

```
public dab.si.EnsembleId getParent()
```

Returns the parent ensemble

getComponentIds

```
public dab.si.ComponentId[] getComponentIds()
```

Returns a reference to ids of the components of the service

isLocalService

```
public boolean isLocalService()
```

Indicates whether the service is local or not

getAccessControlSystem

```
public int getAccessControlSystem()
```

Returns the access control system (cf. DABConstants.acs*)

getLabel

```
public dab.data.Label getLabel()  
    throws DABNotAvailableException
```

Returns the label of the service

Throws:

[DABNotAvailableException](#) - when the label is not available

getLanguage

```
public int getLanguage()  
    throws DABNotAvailableException
```

Returns the language of the service (cf. DABConstants.language*)

Throws:

[DABNotAvailableException](#) - when the language is not available

isPrimaryComponentLanguage

```
public boolean isPrimaryComponentLanguage()  
    throws DABNotAvailableException
```

Indicates, whether the language of the service is the language of the primary component

Throws:

[DABNotAvailableException](#) - when the information is not available

getRegionId

```
public int getRegionId()  
    throws DABNotAvailableException
```

Returns the region id of the service

Throws:

[DABNotAvailableException](#) - when the id is not available

getRegionLabel

```
public dab.data.Label getRegionLabel()  
    throws DABNotAvailableException
```

Returns the region label of the service

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Throws:

[DABNotAvailableException](#) - when the label is not available

getStaticProgrammeType

```
public dab.data.ProgrammeType getStaticProgrammeType()  
                                throws DABNotAvailableException
```

Returns the static programme type

Throws:

[DABNotAvailableException](#) - when the programme type is not available

getDynamicProgrammeType

```
public dab.data.ProgrammeType getDynamicProgrammeType()  
                                throws DABNotAvailableException
```

Returns the dynamic programme type

Throws:

[DABNotAvailableException](#) - when the programme type is not available

getProgrammeNumber

```
public dab.data.ProgrammeNumber getProgrammeNumber()  
                                throws DABNotAvailableException
```

Returns the programme number

Throws:

[DABNotAvailableException](#) - when not available

getTimeOffset

```
public int getTimeOffset()  
           throws DABNotAvailableException
```

Returns the time offset of the service (with respect to the time of the ensemble). The result is returned in minutes. It ranges from -12 hours to 12 hours.

Throws:

[DABNotAvailableException](#) - when the offset is not available

See Also:

[getDate](#)

getAnnouncementSupport

```
public dab.data.AnnouncementSupport getAnnouncementSupport()  
                                    throws DABNotAvailableException
```

Returns the information about announcement support

Throws:

[DABNotAvailableException](#) - when the announcement support is not available

getCountry

```
public int getCountry()
    throws DABNotAvailableException
```

Returns the country information of the service (cf. `DABConstants.country*`)

Throws:

[DABNotAvailableException](#) - when the country information is not available

Class SIId

[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+--dab.si.SIId
```

Direct Known Subclasses:

[ComponentId](#), [EnsembleId](#), [ServiceId](#)

```
abstract public Class SIId
```

```
extends java.lang.Object
```

The SIId is an identifier for a DAB Ensemble, a DAB Service or a DAB Service Component. It defines a handle to one of these service elements and is used to start and stop services or to query service information.

The identifier for each entity is globally unique. This means an identifier for a component or service includes information about the service context as for instance two services are considered different even if they have the same (DAB) service identifier.

Version:

1.07

Constructor Summary		Page
siid ()		391

Method Summary		Page
int	compareTo (<code>dab.si.SIId siid</code>)	391
	This method compares the object with the given object.	
int	compareTo (<code>java.lang.Object object</code>)	391
	This method compares the object with the given object.	
<code>java.lang.String</code>	getId ()	403
	Returns an external representation of the identifier in a textual format.	

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail**SIId**

```
public SIId()
```

Method Detail**compareTo**

```
public int compareTo(java.lang.Object object)
```

This method compares the object with the given object. The behavior is the same as it is specified in the `compareTo` method of the `java.lang.Comparable` interface.

compareTo

```
public int compareTo(dab.si.SIId siid)
```

This method compares the object with the given object. The behavior is the same as it is specified in the `compareTo` method of the `java.lang.Comparable` interface.

getId

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

Returns an external representation of the identifier in a textual format. The returned value can be used to construct a service identifier.

Package dab.xlet

Interface Summary		Page
Xlet	The Xlet is an abstract class that defines the application model used in the dab package.	456
XletContext	The XletContext can be used by the Xlet to communicate with its application controller.	454

Exception Summary		Page
XletStateChangeException	The XletStateChangeException is used to signal errors in the execution of some of the Xlet methods.	454

Interface Xlet

[dab.xlet](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

public Interface **Xlet**

The Xlet is an abstract class that defines the application model used in the dab package.

The `xlet` class contains all the basic methods for controlling and initialising the Xlet component. The method described below are in its essence signals of state changes to the Xlet. This means if the application controller of the Xlet is changing its state, it will call the respective method to indicate the state change to the Xlet.

For a more detailed description of the Xlet model see the java TV API documentation (<http://java.sun.com/products/javatv/>)

Version:

0.2

See Also:

[DABAppProxy](#)

Method Summary		Page
void	destroyXlet (boolean unconditional) The Xlet is requested to stop its operations and to release all the resources, it has allocated.	457
void	initXlet (dab.xlet.XletContext ctx) If this method is called, the Xlet may initialise itself.	456
void	pauseXlet () The Xlet is asked to temporarily freeze its operations.	457
void	startXlet () The Xlet is requested to start its operations.	457

Method Detail

initXlet

```
public void initXlet(dab.xlet.XletContext ctx)
    throws XletStateChangeException
```

If this method is called, the Xlet may initialise itself. The effect is that the Xlet is ready for execution, but has not started its operation, like after a call of `pauseXlet`.

Throws:

[dab.xlet.XletStateChangeException](#) - The exception is thrown if the initialization was not possible.

startXlet

```
public void startXlet()
    throws XletStateChangeException
```

The Xlet is requested to start its operations. In this state the Xlet may hold shared resources. This method will only be called after a call of `initXlet` or `pauseXlet`

Throws:

[dab.xlet.XletStateChangeException](#) - The exception is thrown if the Xlet could not be started.

pauseXlet

```
public void pauseXlet()
```

The Xlet is asked to temporarily freeze its operations. This also means that shared resources are released.

destroyXlet

```
public void destroyXlet(boolean unconditional)
    throws XletStateChangeException
```

The Xlet is requested to stop its operations and to release all the resources, it has allocated. No other methods are called afterwards, if the destroyXlet returns successfully.

Parameters:

`unconditional` - this parameter indicates that any requests of the Xlet not to be destroyed are ignored (if set to true) or not (if set to false).

Throws:

[dab.xlet.XletStateChangeException](#) - The exception is thrown if the Xlet (or its depending resources) could not be destroyed or if it requests not to be destroyed. If the request is accepted, destroyXlet will be called again at a later time.

Interface XletContext

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

[dab.xlet](#)

```
public Interface XletContext
```

The XletContext can be used by the Xlet to communicate with its application controller.

Version:

0.2

Method Summary		Page
void	destroyed () This method indicates to the application controller that the Xlet has destroyed itself in the same way as if destroyXlet has been called	455
java.lang.Object	getXletProperty (java.lang.String key) Requests a property from the environment.	455
void	paused () The Xlet indicates that it has paused its activities in the same manner as if pauseXlet has been called.	455
void	resumeRequest () If the Xlet has paused its activities, it may use this method to signal the application controller to start the Xlet again.	455

Method Detail**destroyed**

```
public void destroyed()
```

This method indicates to the application controller that the Xlet has destroyed itself in the same way as if `destroyXlet` has been called

See Also:

[destroyXlet](#)

getXletProperty

```
public java.lang.Object getXletProperty(java.lang.String key)
```

Requests a property from the environment. The value of the property with the given key is returned or is NULL if no such property exists.

paused

```
public void paused()
```

The Xlet indicates that it has paused its activities in the same manner as if `pauseXlet` has been called.

See Also:

[pauseXlet](#)

resumeRequest

```
public void resumeRequest()
```

If the Xlet has paused its activities, it may use this method to signal the application controller to start the Xlet again. Note, it is necessary to use a different thread in the application controller for calling `startXlet` to avoid deadlocks.

Class XletStateChangeException

[dab.xlet](#)

```
java.lang.Object
|
+--java.lang.Throwable
    |
    +--java.lang.Exception
        |
        +--dab.xlet.XletStateChangeException
```

All Implemented Interfaces:

java.io.Serializable

```
public Class XletStateChangeException
```

```
extends java.lang.Exception
```

The *XletStateChangeException* is used to signal errors in the execution of some of the *Xlet* methods.

Version:
0.2

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`, `getStackTrace`,
`initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`,
`toString`

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Class `ResourceConflictException`

[dab](#)

`java.lang.Object`

|

+-- `java.lang.Throwable`

|

+-- `java.lang.Exception`

|

+-- [dab.DABException](#)

|

+-- **`dab.ResourceConflictException`**

All Implemented Interfaces:

`java.io.Serializable`

public Class **`ResourceConflictException`**

extends [DABException](#)

The exception indicates unsolved resource conflicts.

Version:
0.2

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`, `getStackTrace`,

```
initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace,
toString
```

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Interface DABSource

[dab](#)

All Known Implementing Classes:

[DABClient](#)

public Interface **DABSource**

DABSource defines the interface of a DAB resource (usually a DAB receiver). The interface is asynchronous. When an application issues a requests, it gets back the results as confirmation and notification events. Look at the particular methods for more details.

Version:

1.07

See Also:

[DABListener](#)

Method Summary		Page
void	getComponentInfoReq (dab.si.ComponentId componentId) The getComponentInfoReq request initiates a GetComponentInfo command.	338
void	getEnsembleInfoReq (dab.si.EnsembleId id) The getEnsembleInfoReq method initiates a GetEnsembleInfo command.	338
void	getLocationInfoReq (int type, int mode, int desiredDelta, int desiredAccuracy) The getLocationInfoReq initiates the GetLocationInfoCommand.	340
void	getServiceInfoReq (dab.si.ServiceId id) The getServiceInfoReq requests initiates a GetServiceInfo command.	338
void	operationControlReq (int attribute, java.lang.Object value) The OperationControl command enables the DAB client to change or read receiver parameters.	340
void	respondConflictResolutionReq (int transaction, int turn, int operation, int suboperation, int answer) The respondConflictResolutionReq is used to respond to a resource conflict notification.	340
void	scanReq (int searchMode, int tables, int startFrequency, int stopFrequency, int tansmissionModes, int notifications) The ScanReq request initiates a Scan command.	338

void	searchReq (int searchMode, int tables, int startFrequency, int stopFrequency, int transmissionModes, int notifications) The searchReq request initiates a Search command.	337
void	selectApplicationReq (dab.si.ComponentId serviceId, dab.data.ObjectId objectId) The selectApplicationReq selects applications The SelectApplication command enables a client to load and control an application.	339
void	selectComponentReq (dab.si.ComponentId id, int selectionMode) The selectComponentReq request initiates the SelectComponent command.	339
void	selectComponentStreamReq (dab.si.ComponentId componentId) The following request provides access to the DAB transport streams.	339
void	selectObjectReq (dab.si.ComponentId id, dab.data.ObjectId objectId, int requestMode, boolean replaceSelections, int deliveryMode, int cacheHint) The selectObjectReq request initiates the SelectObject command.	339
void	selectReceptionInfoReq (boolean synchronizationNotification, boolean bitErrorRateNotifcations, boolean muteStateNotifications, boolean requestOnce) The selectReceptionInfoReq request initiates the SelectReceptionInfo command.	339
void	selectSIRReq (boolean ensembleInfo, boolean serviceInfo, boolean componentInfo, boolean autoDelivery) The selectSIRReq method initiates a SelectSI command.	338
void	tuneReq (int tuneFrequency, int transmissionMode) The tuneReq request initiates the Tune command.	337

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

Method Detail

tuneReq

```
public void tuneReq(int tuneFrequency,
                    int transmissionMode)
    throws DABException,
           java.lang.SecurityException
```

The tuneReq request initiates the Tune command. The Tune command sets directly a specified DAB frequency. A DAB receiver must be tuned to a DAB frequency and synchronized in order to get access to DAB services. A tuned DAB receiver tries automatically to synchronize on a DAB Ensemble.

The Tune command is used to select a specified DAB frequency. The tuneReq request initiates the Tune command. Depending on the specification for the Transmissionmode it is tested if a DAB Ensemble can be detected. If the connected DAB receiver supports automatic detection the default setting for transmissionMode (=DABConstants.transmissionModeAutomatic) can be used. Otherwise it has to be specified which transmission modes should be tested. The result of the command is delivered by the tuneCnf confirmation. All currently existing selections of audio and data services or selections of data objects are automatically stopped before tuning is performed by the DAB receiver.

Parameters:

`tuneFrequency` - This parameter specifies the frequency the DAB receiver will be tuned to in Hertz.

`transmissionModes` - This parameter specifies the transmission modes a DAB receiver tests for DAB Ensembles. The default value is `DABConstants.transmissionModeAutomatic` which means that the receiver is automatically detecting the transmission mode. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.transmissionModeAutomatic`: The transmission mode is automatically detected. All other flags are ignored in this case.
- `DABConstants.transmissionMode1`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 1.
- `DABConstants.transmissionMode2`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 2.
- `DABConstants.transmissionMode3`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 3.
- `DABConstants.transmissionMode4`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 4.

Throws:

[DABException](#) - in case of invalid DAB parameter

[SecurityException](#) - unauthorized usage of `tuneReq`

See Also:

[tuneCnf](#), [searchReq](#), [searchCnf](#), [searchNtf](#)

searchReq

```
public void searchReq(int searchMode,
                      int tables,
                      int startFrequency,
                      int stopFrequency,
                      int transmissionModes,
                      int notifications)
    throws DABException,
           java.lang.SecurityException
```

The `searchReq` request initiates a Search command. The Search command searches for a DAB Ensemble according to a specified search mode. After a successful execution of the Search command a DAB Ensemble has been found, the state Tuned is entered and the DAB receiver tries to synchronize automatically to the found DAB Ensemble.

The Search command is used to search for a DAB Ensemble. The `searchReq` request initiates the search and specifies the frequencies and transmission modes to test. Additionally the notifications can be specified which the DAB client gets while the command is executed. Searching for an ensemble may require a substantial amount of time from only a second up to several minutes. This depends also on the search mode specified. If the reception conditions are bad it is possible that no DAB Ensemble at all is detected. In order to stop searching for a DAB Ensemble the Tune command can be used which tunes the DAB Receiver to a certain frequency independent from the reception conditions. The start of searching is indicated by a `SearchNtf` event with a status code 'Started'. In this case the state machine of Tune State enters the state Searching (see Figure 4). In case that the previous state was Tuned all currently existing selections of services or objects are stopped automatically. While searching is performed several notifications delivering information about the current status are sent to the client. The command ends with a `SearchCnf` event.

Parameters:

`searchMode` - This parameter specifies the way the DAB receiver is searching for a DAB Ensemble. The default value is `searchSearchAutomatic` which means it is searching according to a default method. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.SearchModeAutomatic`: default method

- `DABConstants.SearchMode16kHzSteps`: The frequency range is searched in 16 kHz steps. This is a very intensive search which means that command execution can take a large amount of time.
- `DABConstants.SearchModeUp`: The search direction is from low to high frequencies.
- `DABConstants.SearchModeDown`: The search direction is from high to low frequencies.
- `DABConstants.SearchModeUseTables`: The search is based on the specified frequency tables.
- `DABConstants.SearchModeUseFrequencyRange`: The search is based on the specified frequency range.
- `DABConstants.SearchModeContinuous`: The search is looping over the specified frequency range until a DAB Ensemble has been found. The default is to stop after the specified frequency range has been checked once.

`tables` - This parameter specifies frequency tables the receiver uses in order to search for DAB Ensembles. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.searchCEPTFrequencyTableBandIII`: The frequencies according to the CEPT frequency table for Band III are tested for DAB Ensembles.
- `DABConstants.SearchCEPTFrequencyTableLBand`: The frequencies according to the CEPT L-Band table are tested for DAB Ensembles.
- `DABConstants.SearchCanadaFrequencyTableLBand`: The frequencies according to the Canadian L-Band table are tested for DAB Ensembles.

`transmissionModes` - This parameter specifies the transmission modes a DAB receiver tests for DAB Ensembles. The default value is `DABConstants.transmissionModeAutomatic` which means that the receiver is automatically detecting the transmission mode. The parameter is a flag field supporting the following flags which can be specified together:

- `transmissionModeAutomatic`: The transmission mode is automatically detected. All other flags are ignored.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

- `transmissionMode1`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 1.
- `transmissionMode2`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 2.
- `transmissionMode3`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 3.
- `transmissionMode4`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 4.

`notifications` - This parameter specifies the type of notifications the client wants to get while the `Seek` command is performed. The parameter is a flag field supporting the following flags which can be specified together:

- `notificationsOff`: No intermediate notifications are sent. Only a `SearchNtf` notification which informs about the start of searching is sent.
- `notifications16kHzSteps`: With each 16 kHz step a notification is sent. This is only possible if 16kHz step searching is specified as search mode.
- `notificationsTableEntry`: With each table entry frequency a notification is sent. This is the default value.

See Also:

[searchCnf](#), [searchNtf](#), [tuneReq](#), [tuneCnf](#)

scanReq

```
public void scanReq(int searchMode,
                    int tables,
                    int startFrequency,
                    int stopFrequency,
                    int transmissionModes,
                    int notifications)
    throws DABException,
           java.lang.SecurityException
```

The `ScanReq` request initiates a Scan command. The Scan command is used in order to perform a search for all available DAB Ensembles in a specified frequency range. Depending on the frequency range and the search mode this operation may require a substantial amount of time from only a second up to several minutes. The command is started by the `ScanReq` request and is finished with the `ScanCnf` confirmation. In between `ScanNtf` notification are sent in order to inform about the current status of scanning if notifications are requested.

In case of searching from lower to higher frequencies (`searchMode=DABConstants.searchModeUp`) the value of `startFrequency` is not allowed to be larger than the value of `stopFrequency`. In case of searching from higher to lower frequencies (`searchMode=DABConstants.searchModeDown`) the value of `startFrequency` is not allowed to be smaller than the value of `stopFrequency`.

Parameters:

`searchMode` - This parameter specifies the way the DAB Receiver is searching for a DAB Ensemble. The default value is `DABConstants.searchModeAutomatic` which means it is searching according to a default method. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.searchModeAutomatic`: default method
- `DABConstants.searchMode16kHzSteps`: The frequency range is searched in 16 kHz steps.
- `DABConstants.searchModeUp`: The search direction is from low to high frequencies.
- `DABConstants.searchModeDown`: The search direction is from high to low frequencies.
- `DABConstants.searchModeUseTables`: The search is based on the specified frequency tables.
- `DABConstants.searchModeUseFrequencyRange`: The search is based on the specified frequency range.
- `DABConstants.searchModeContinuous`: The search is looping over the specified frequency range until a DAB Ensemble has been found. The default is to stop after the specified frequency range has been checked once.

`tables` - This parameter specifies frequency tables the receiver uses in order to search for DAB Ensembles. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.searchCEPTFrequencyTableBandIII`: The frequencies according to the CEPT frequency table for Band III are tested for DAB Ensembles.
- `DABConstants.searchCEPTFrequencyTableLBand`: The frequencies according to the CEPT L-Band table are tested for DAB Ensembles.
- `DABConstants.searchCanadaFrequencyTableLBand`: The frequencies according to the Canadian L-Band table are tested for DAB Ensembles.

`startFrequency` - This parameter specifies the start frequency at which the DAB Receiver starts its search for DAB Ensembles.

`stopFrequency` - This parameter specifies the stop frequency at which the DAB Receiver stops its search for DAB Ensembles.

`transmissionModes` - This parameter specifies the transmission modes a DAB Receiver should look for DAB Ensembles. The default value is `DABConstants.transmissionModeAutomatic` which means that the receiver is automatically detecting the transmission mode. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.transmissionModeAutomatic`: The transmission mode is automatically detected.
- `DABConstants.transmissionMode1`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 1.
- `DABConstants.transmissionMode2`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 2.
- `DABConstants.transmissionMode3`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 3.
- `DABConstants.transmissionMode4`: At the specified frequency it is tested if a DAB Ensemble is sent in transmission mode 4.

`notifications` - This parameter specifies the type of notifications wanted by the application while the Seek command is performed. The parameter is a flag field supporting the following flags which can be specified together:

- `DABConstants.notificationsOff`: No notifications are sent.
- `DABConstants.notifications16kHzSteps`: With each 16 kHz step a notification is sent.
- `DABConstants.notificationsTableEntry`: With each table entry frequency a notification is sent. This is the default value.

See Also:

[scanCnf](#), [scanNtf](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

selectSIReq

```
public void selectSIReq(boolean ensembleInfo,  
                        boolean serviceInfo,  
                        boolean componentInfo,  
                        boolean autoDelivery)  
    throws DABException,  
           java.lang.SecurityException
```

The `selectSIReq` method initiates a `SelectSI` command. The `SelectSI` command starts, stops or changes subscription to Service Directory Information.

The `SelectSI` command allows a DAB client to subscribe for Service Directory Information. The Service Directory contains all available ensembles, services, components and related information. The subscription is requested by the `selectSIReq` request and is confirmed with the `selectSICnf` confirmation. The subscription level can be changed by another `SelectSI` command. This includes the termination of subscription.

After a successful subscription a connected client receives `SINtf` notifications. Just after the subscription has been activated the complete content of the Service Directory is mapped on `SINtf` notifications. This means for each stored instance of a service element (ensemble, service and component) is a `SINtf` notification sent which indicates that this service element is available (Added). As time goes on `SINtf` notifications are sent which indicate that a new service element is available (Added), that an existing is no longer available (Removed) or that its attributes have changed (Changed).

By use of `autoDelivery` it can be specified if `SINtf` sends only a notification or a notification together with the related information object. If subscription is terminated by setting `ensembleInfo`, `serviceInfo` and `componentInfo` to false, then `autoDelivery` has no meaning.

By selecting a certain subscription level the client is informed about all currently known service elements by sending related `SINtf` notifications. This means if a client subscribes for service-specific notifications and seven services exist at this time, then seven `SINtf(DABConstants.serviceAdded)` notifications are generated. The client is not informed about known ensembles or components. As time goes on the client is informed when new services are added, known services are removed or changed. If a currently selected subscription level is increased meaning that more notification types are subscribed then the client is informed about all currently known service elements that are related to the new subscribed notification type. This means if a subscription is changed from service-specific to service-specific and component-specific change notifications, then for each currently known Component a `SINtf(DABConstants.componentAdded)` notification is generated.

As time goes on the client is informed when new services or components are added, known services or components are removed or changed. If a currently selected subscription level is decreased meaning that less notification types are subscribed then the client is informed only about notifications related to the remaining subscribed notification types. This means if a subscription is changed from service-specific and component-specific to service-specific notifications, then the client is informed when new services are added, known services are removed or changed. But the client is no longer notified about changes related to components.

Parameters:

`ensembleInfo` - This parameter specifies if ensemble-specific notifications will be sent to the DAB client. The following values are supported:

- `true`: The DAB client is notified about `DABConstants.ensembleAdded`, `DABConstants.ensembleChanged` and `DABConstants.ensembleRemoved` events. This is the default setting.
- `false`: The DAB client is not notified about `DABConstants.ensembleAdded`, `DABConstants.ensembleChanged` and `DABConstants.ensembleRemoved` events.

`serviceInfo` - this parameter specifies if service-specific notifications will be sent to the DAB client. The following values are supported:

- `true`: The DAB client is notified about `DABConstants.serviceAdded`, `DABConstants.serviceChanged` and `DABConstants.serviceRemoved` events. This is the default setting.
- `false`: The DAB client is not notified about `DABConstants.serviceAdded`, `DABConstants.serviceChanged` and `DABConstants.serviceRemoved` events.

`componentInfo` - This parameter specifies if component-specific notifications will be sent to the DAB client. The following values are supported:

- true: The DAB client is notified about `DABConstants.componentAdded`, `DABConstants.componentChanged` and `DABConstants.componentRemoved` events. This is the default setting.
- false: The DAB client is not notified about `DABConstants.componentAdded`, `DABConstants.componentChanged` and `DABConstants.componentRemoved` events.

`autoDelivery` - This parameter specifies if the information related to the notification is sent together with the notification (`SINtf`) or not. The following values are supported:

- true: The `SINtf` notification delivers the notification together with the information object. The information object is sent together with the notification if the notification type is -Added or -Changed. In case of -Removed no information object is sent because it is no longer existing. This is the default setting.
- false: The `SINtf` notification delivers only the notification. The information object (`EnsembleInfo`, `ServiceInfo` or `ComponentInfo`) itself can be obtained by use of `getEnsembleInfoReq`, `getServiceInfoReq` or `getComponentInfoReq`.

See Also:

[selectSICnf](#), [siNtf](#)

getEnsembleInfoReq

```
public void getEnsembleInfoReq(dab.si.EnsembleId id)
    throws DABException,
           java.lang.SecurityException
```

The `getEnsembleInfoReq` method initiates a `GetEnsembleInfo` command. The `GetEnsembleInfo` command requests information about the specified DAB Ensemble.

The `GetEnsembleInfo` command provides a DAB client with information about a specified DAB Ensemble, e.g. Label, No of Services, and so on. The command is initiated by a `getEnsembleInfoReq` request and is finished by a `getEnsembleInfoCnf` confirmation.

Parameters:

`id` - This parameter is a handle identifying the DAB Ensemble.

See Also:

[getEnsembleInfoCnf](#)

getServiceInfoReq

```
public void getServiceInfoReq(dab.si.ServiceId id)
    throws DABException,
           java.lang.SecurityException
```

The `getServiceInfoReq` requests initiates a `GetServiceInfo` command. The `GetServiceInfo` command requests information about a specified DAB Service.

The `GetServiceInfo` command provides a DAB client with information about a specified DAB Service, e.g. Label, No of Components, and so on. The command is initiated by a `getServiceInfoReq` request and is finished by a `getServiceInfoCnf` confirmation.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Parameters:

`id` - This parameter is a handle identifying the DAB Service.

See Also:[getServiceInfoCnf](#)**getComponentInfoReq**

```
public void getComponentInfoReq(dab.si.ComponentId componentId)
    throws DABException,
           java.lang.SecurityException
```

The `getComponentInfoReq` request initiates a `GetComponentInfo` command. The `GetComponentInfo` command requests information about a specified DAB Component.

The `GetComponentInfo` command provides a DAB client with information about a specified DAB Component, e.g. Label, Language and so on. The command is initiated calling `getComponentInfoReq` and is finished by a call to `getComponentInfoCnf`.

Parameters:

`serviceId` - This parameter is a handle identifying the DAB Component.

See Also:[getComponentInfoCnf](#)**selectReceptionInfoReq**

```
public void selectReceptionInfoReq(boolean synchronizationNotification,
    boolean bitErrorRateNotifcations,
    boolean muteStateNotifications,
    boolean requestOnce)
    throws DABException,
           java.lang.SecurityException
```

The `selectReceptionInfoReq` request initiates the `SelectReceptionInfo` command. The `SelectReceptionInfo` command starts, stops or changes subscription to state change notifications concerning reception conditions. It is possible to monitor synchronization, biterrrorrate and audio decoder muting.

The `SelectReceptionInfo` command allows a DAB client to subscribe for state change notifications concerning reception conditions in terms of synchronization, biterrrorrate and audio decoder muting. The subscription is requested by the `selectReceptionInfoReq` request and is confirmed with the `selectReceptionInfoCnf` confirmation. The subscription level can be changed by another `SelectReceptionInfo` command. This includes stopping of subscription. After a successful subscription the calling DAB client receives `ReceptionInfoNtf` notifications when state changes occur.

Parameters:

`synchronizationNotification` - This parameter specifies if the calling client is notified about state changes concerning DAB signal synchronization. If the parameter is set to true (default) notifications are provided, if it is set to false no notifications are provided.

`bitErrorRateNotifcations` - This parameter specifies if the calling client is notified about state changes concerning the biterrrorrate. If the parameter is set to true (default) notifications are provided, if it is set to false no notifications are provided.

`muteStateNotifications` - This parameter specifies if the calling client is notified about state changes concerning the mute state of the audio decoder. If the parameter is set to true (default) notifications are provided, if it is set to false no notifications are provided.

`requestOnce` - This parameter specifies if the reception condition information is wanted only once. In this case the reception condition is once detected and the DAB client informed by one and only one `receptionInfoNtf` call.

See Also:[selectReceptionInfoCnf](#), [receptionInfoNtf](#)**selectComponentReq**

```
public void selectComponentReq(dab.si.ComponentId id,
    int selectionMode)
    throws DABException,
           java.lang.SecurityException
```

The `selectComponentReq` request initiates the `SelectComponent` command. The `SelectComponent` command starts or stops an application delivered in a DAB Component.

The `SelectComponent` command allows to start or stop applications delivered in DAB components. In general more than one component of the same DAB Ensemble can be selected simultaneously. It is possible to select one audio component, all programme-associated data components of the selected audio component and more than one independent data component at the same time. The selection of a component is requested by the `selectComponentReq` request and is confirmed by the `selectComponentCnf` confirmation. It is possible that a component is removed from a DAB Ensemble which means it is no longer broadcast and therefore no longer available. This is indicated by a `SINtf` call and means that the selection is removed automatically. If the selection of a component is removed also all existing object selections belonging to the component are removed.

If the user application is a slide show or a dynamic label, its objects are delivered automatically (using `objectNtf` notifications) after the `SelectComponent` confirmation was sent.

If the selected component is an audio service, its PAD data services become available as "virtual components". This means service information is generated for all PAD services and they can be selected. If the selection of the audio service is stopped, also all PAD services are stopped and they are not available anymore.

If the component is not in the current ensemble, it depends on the implementation whether it is selected nevertheless.

Parameters:

`id` - This parameter is a pointer to the identifier of the DAB Component which is to be selected. If all component selections should be removed (set `selectionMode` to `DABConstants.selectionModeRemoveAll`) this parameter is ignored and should be set to null.

`selectionMode` - This parameter specifies the selection mode for the component. The following flags are supported:

- `DABConstants.selectionModeReplace`: All currently selected components of the same type are stopped and the specified component is to be started. The same type means an audio component replaces any other selected audio component, a data component replaces all other selected independent data components and a programme-associated data component replaces all other selected programme-associated data components.
- `DABConstants.selectionModeAdd`: The application delivered by the specified component is to be started. Other selected components are not affected.
- `DABConstants.selectionModeRemove`: The selection of the specified component is stopped.
- `DABConstants.selectionModeRemoveAll`: All existing component selections are removed. Set `serviceld` to null in this case.

See Also:

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

[selectComponentCnf](#), [siNtf](#), [serviceObjectReq](#)

selectComponentStreamReq

```
public void selectComponentStreamReq(dab.si.ComponentId componentId)
    throws DABException,
    java.lang.SecurityException
```

The following request provides access to the DAB transport streams. The requested stream is delivered back in the confirmation, which ends the command.

Parameters:

`componentId` - the service identifier of the component which carries the stream

See Also:[selectComponentStreamCnf](#)**selectApplicationReq**

```
public void selectApplicationReq(dab.si.ComponentId serviceId,  
                                dab.data.ObjectId objectId)  
    throws DABException,  
           java.lang.SecurityException
```

The selectApplicationReq selects applications

The SelectApplication command enables a client to load and control an application. The request is confirmed with the selectApplicationCnf confirmation.

Parameters:

serviceId - the component in which the application is located
objectId - the id of the start object

See Also:[selectApplicationCnf](#)**selectObjectReq**

```
public void selectObjectReq(dab.si.ComponentId id,  
                             dab.data.ObjectId objectId,  
                             int requestMode,  
                             boolean replaceSelections,  
                             int deliveryMode,  
                             int cacheHint)  
    throws DABException,  
           java.lang.SecurityException
```

The selectObjectReq request initiates the SelectObject command. The SelectObject command selects an object from a selected DAB Component. This includes requesting an object from a data component, delivery after reception and notification of updates as long as the object is selected.

The SelectObject command selects an object from a selected component. Selection means it is requested for delivery and if wanted also updates of the object are delivered. Additionally it is possible to give some hints for caching. More than one object and also from more than one component can be selected simultaneously. The selection of an object is requested by the selectObjectReq request and is confirmed by the selectObjectCnf confirmation. The object is delivered using the objectNtf method. This includes first-time delivery and all updates. Beyond starting or stopping a selection it is possible to remove all other selections belonging to the same component by setting parameter replaceSelections to true. It is possible that a component is removed from a DAB Ensemble. This is indicated by a serviceInfoNtf call. In this case also the selected objects of the service are no longer selected.

It is possible that an object is removed from current on-air service. This is indicated by an objectNtf call. In this case the selections for this object are automatically disabled.

Currently object selection makes only sense with applications of type BroadcastWebSite. Objects of applications like Slideshows or Dynamic Label are delivered automatically by objectNtf calls.

Parameters:

id - This parameter identifies the selected component the object is belonging to.
objectId - This parameter identifies the object which is to be selected.
selectionMode - This parameter specifies the selection mode of the object. The following values are supported:

- DABConstants.requestModeOff: This is used in order to stop the selection of objects which are requested with the request mode DABConstants.requestModeUpdate. It is not needed for objects which are requested with the DABConstants.requestModeOnce flag except for the case that a SelectObjectReq is pending and the delivery is no longer wanted.

- `DABConstants.requestModeOnce`: The object is requested for one-time delivery. After the first reception from the broadcast channel the object is delivered to the connected DAB client. The client is not notified about new versions.
- `DABConstants.requestModeUpdate`: The object is requested for update delivery. After the first reception from the broadcast channel the object is delivered to the connected client. Additionally each new version of the object is delivered.

`replaceSelections` - This parameter specifies if all current object selections belonging to the component identified by `serviceld` are replaced with this selection. If this parameter is set to true, then all selections are removed. If this parameter is set to false, then existing selections remain unchanged.

`deliveryMode` - This parameter specifies the delivery mode of the object. The following values are supported:

- `DABConstants.deliveryModeComplete`: Only the complete object is delivered to the DAB client.
- `DABConstants.deliveryModePartial`: The object may be delivered in parts.

`cacheHint` - This parameter specifies a hint for caching of the selected object.

See Also:

[selectObjectCnf](#), [objectNtf](#)

getLocationInfoReq

```
public void getLocationInfoReq(int type,
                               int mode,
                               int desiredDelta,
                               int desiredAccuracy)
    throws DABException,
           java.lang.SecurityException
```

The `getLocationInfoReq` initiates the `GetLocationInfoCommand`.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Parameters:

`type` - This parameter indicates the type of location information, that is requested. Supported flags are `DABConstants.LocationInfoPosition` and `DABConstants.LocationInfoRegionId`.

`mode` - The information is delivered according to the values of this parameter:

- `DABConstants.LocationInfoOnce`: the information is delivered only one time. The parameter `desiredDelta` is not considered.
- `DABConstants.LocationInfoPeriodByTime`: The information is delivered in intervals given by the value of `desiredDelta` (in milliseconds)
- `DABConstants.LocationInfoPeriodByDistance`: The information is delivered after the distance has passed given by the value of `desiredDelta` (in meters)
- `DABConstants.LocationInfoStop`: The delivery of information is stopped. The parameter `desiredDelta` is not considered.

`desiredDelta` - cf. description of the mode parameter

`desiredAccuracy` - This parameter indicates the desired accuracy in meters. The value is only considered if `type&DABConstants.LocationInfoPosition!=0`

This command is optional and may only partially be supported (e.g. only mode=DABConstants.LocationInfoOnce and mode=DABConstants.LocationInfoStop) or may not be supported at all.

See Also:

[getLocationInfoCnf](#), [locationInfoNtf](#)

respondConflictResolutionReq

```
public void respondConflictResolutionReq(int transaction,
                                         int turn,
                                         int operation,
                                         int suboperation,
                                         int answer)
```

The respondConflictResolutionReq is used to respond to a resource conflict notification.

Parameters:

transaction - the identifier of the transaction of the resource conflict
turn - the code of the turn (cf. DABConstants.conflictResolutionTurn*)
operation - the code of the operation (cf. DABConstants.conflictResolutionOperation*)
suboperation - the code of the suboperation (cf. DABConstants.conflictResolutionSuboperation*)
answer - the actual answer (cf. DABConstants.conflictResolutionAnswer*)

See Also:

[conflictResolutionNtf](#), [respondConflictResolutionCnf](#)

operationControlReq

```
public void operationControlReq(int attribute,
                                 java.lang.Object value)
    throws DABException,
           java.lang.SecurityException
```

The OperationControl command enables the DAB client to change or read receiver parameters. The command is initiated by operationControlReq and is finalized by the confirmation operationControlReq.

Parameters:

attribute - The parameter can be set as follows:

- DABConstants.operationControlSetVolume: The volume of the receiver is set. The parameter value has to be of type Integer in the range from 0 to 100 (percent).
- DABConstants.operationControlGetVolume: The volume of the receiver is read. The parameter value is not considered.
- DABConstants.operationControlSetServiceFollowing: The service following feature is changed. Value has to be of type Boolean. If it is set to true and the receiver supports service following, then service following for services is switched on. If it is set to false, service following is switched off.
- DABConstants.operationControlGetServiceFollowing: Read the state of the service following. The parameter value is not considered.
- DABConstants.operationControlGetServiceFollowingNotifications: Instruct the package to send service following notifications. If the parameter value (of type Boolean) is set to true, the notifications are sent. If it is set to false, then no further notifications are sent.
- DABConstants.operationControlSetDRCMode: Sets the DRC (Dynamic range control) mode. The DAB concept provides the option of Dynamic Range Control (DRC). The information is generated from the broadcaster's side (transported inside PAD, Programme Associated Data) to influence the audio output signal's dynamic range. The audio output signal will be modified if the option is activated with this call.

Value has to be of type Boolean. If it is set to true and the receiver supports DRC, then the DRC mode for audio services is switched on. If it is set to false, the DRC mode is switched off.

- `DABConstants.operationControlGetDRCMode`: Read the state of the DRC mode. The parameter value is not considered.
- `DABConstants.operationControlGetDRCModeNotifications`: Instruct the package to send DRC mode notifications. If the parameter value (of type Boolean) is set to true, the notifications are sent. If it is set to false, then no further notifications are sent.

value - cf. description of attribute

See Also:

[operationControlCnf](#), [serviceFollowingNtf](#), [drcModeNtf](#)

Class **DABReceiverAddress**

[dab](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

`java.lang.Object`

|

+--**dab.DABReceiverAddress**

public Class **DABReceiverAddress**

extends `java.lang.Object`

DABReceiverAddress is used to specify the location of DAB receivers.

Version:

1.0

Constructor Summary	Page
DABReceiverAddress (<code>java.lang.String address</code>) Generates a DABReceiverAddress object from a textual representation of the address.	276

Method Summary	Page
<code>java.lang.String</code> getAddress () Generates a textual representation of the object, that can be used to construct a DABReceiverAdress object	276
<code>java.lang.String</code> toString () Generates a textual representation of the object.	389

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

DABReceiverAddress

```
public DABReceiverAddress(java.lang.String address)
```

Generates a *DABReceiverAddress* object from a textual representation of the address. Note, that the actual format of address depends on the implementation

See Also:

[getAddress](#)

Method Detail

toString

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

Generates a textual representation of the object.

Overrides:

toString in class java.lang.Object

getAddress

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

Generates a textual representation of the object, that can be used to construct a *DABReceiverAdress* object

Class DABNotAvailableException

[dab](#)

```
java.lang.Object
```

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

```
|  
+--java.lang.Throwable  
    |  
    +--java.lang.Exception  
        |  
        +--dab.DABException  
            |  
            +--dab.DABNotAvailableException
```

All Implemented Interfaces:

java.io.Serializable

public Class **DABNotAvailableException**

extends [DABException](#)

The *DABNotAvailableException* is thrown when particular data is currently not available or even not at all available. This usually happens with respect to so-called optional attributes.

Version:

1.0

Constructor Summary	Page
DABNotAvailableException ()	277

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

DABNotAvailableException

public **DABNotAvailableException** ()

Interface DABListener

[dab](#)

All Known Implementing Classes:

[DABAdapter](#)

public Interface **DABListener**

DABListener defines the interface for DAB listeners.

Note: to support J2ME MIDP profile DABListener does not implement java.util.EventListener

Version:

1.07

Method Summary	Page
void componentNtf (dab.events.ComponentNtfEvent e)	349
The componentNtf method is called if there are changes to the selection mode of a component.	

void	<u>conflictResolutionNtf</u> (dab.events.ConflictResolutionNtfEvent e) The method is called for notifying the listener of resource conflicts.	350
void	<u>drcModeNtf</u> (dab.events.DRCModeNtfEvent e) The notification informs about DRC mode changes.	350
void	<u>GetComponentInfoCnf</u> (dab.events.GetComponentInfoCnfEvent e) The GetComponentInfoCnf message finalizes the GetComponentInfo command and delivers information about a requested DAB Component to a DAB client.	348
void	<u>GetEnsembleInfoCnf</u> (dab.events.GetEnsembleInfoCnfEvent e) The GetEnsembleInfoCnf method finalizes the GetEnsembleInfo command and delivers information about a requested DAB Ensemble to a DAB client.	348
void	<u>getLocationInfoCnf</u> (dab.events.GetLocationInfoCnfEvent e) getLocationInfoCnf confirms the getLocationInfo command.	350
void	<u>getServiceInfoCnf</u> (dab.events.GetServiceInfoCnfEvent e) A call to the getServiceInfoCnf method finalizes the GetServiceInfo command and delivers information about a requested DAB Service to a DAB client.	348
void	<u>locationInfoNtf</u> (dab.events.LocationInfoNtfEvent e) locationInfoNtf notifies about location information.	350
void	<u>objectNtf</u> (dab.events.ObjectNtfEvent e) The objectNtf method is called as a consequence of selecting objects from a data component by use of the SelectObject command.	349
void	<u>operationControlCnf</u> (dab.events.OperationControlCnfEvent e) The confirmation indicates the result of the operationControl command.	350
void	<u>receptionInfoNtf</u> (dab.events.ReceptionInfoNtfEvent e) The receptionInfoNtf method is called as a consequence of subscribing to state changes in synchronization, biterrorrate and audio decoder muting.	349
void	<u>respondConflictResolutionCnf</u> (dab.events.RespondConflictResolutionCnfEvent e) The method is called for confirming a reaction to a resource conflict.	350
void	<u>scanCnf</u> (dab.events.ScanCnfEvent e) The ScanCnf message finalizes a Scan command.	347
void	<u>scanNtf</u> (dab.events.ScanNtfEvent e) The ScanNtf message is sent after a search for all available DAB Ensembles in a specified frequency range is started by the ScanReq message.	347
void	<u>searchCnf</u> (dab.events.SearchCnfEvent e) The searchCnf method finalizes a Search command and provides information about the command status, currently selected DAB frequency and current reception conditions.	347
void	<u>searchNtf</u> (dab.events.SearchNtfEvent e) A SearchNtf event is sent after a search for a DAB Ensemble was started searchReq.	347

void	<u>selectApplicationCnf</u> (dab.events.SelectApplicationCnfEvent e) The method is called as a consequence of selecting an application from a data component by use of the SelectApplication command.	349
void	<u>selectComponentCnf</u> (dab.events.SelectComponentCnfEvent e) The SelectComponentCnf confirmation finalizes the SelectComponent command.	349
void	<u>selectComponentStreamCnf</u> (dab.events.SelectComponentStreamCnfEvent e) The selectComponentStreamCnf method returns the requested stream (if the command was succesful) and informs about the result of the command.	349
void	<u>selectObjectCnf</u> (dab.events.SelectObjectCnfEvent e) The SelectObjectCnf method finalizes the SelectObject command.	349
void	<u>selectReceptionInfoCnf</u> (dab.events.SelectReceptionInfoCnfEvent e) The selectReceptionInfoCnf method finalizes the SelectReceptionInfo command.	348
void	<u>selectSICnf</u> (dab.events.SelectSICnfEvent e) The selectSICnf method finalizes a SelectSI command and indicates current settings.	348
void	<u>serviceFollowingNtf</u> (dab.events.ServiceFollowingNtfEvent e) The notification informs about service following actions.	350
void	<u>siNtf</u> (dab.events.SINtfEvent e) The siNtf notication is sent as a consequence of subscribing to Service Directory Information.	348
void	<u>systemFailureNtf</u> (dab.events.SystemFailureNtfEvent e) SystemFailureNtf notifies about severe problems with the hardware or the middleware (e.g.	351
void	<u>tuneCnf</u> (dab.events.TuneCnfEvent e) The TuneCnf method finalizes a Tune command and is sent as a response to a TuneReq message.	347

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Detail

tuneCnf

```
public void tuneCnf(dab.events.TuneCnfEvent e)
```

The TuneCnf method finalizes a Tune command and is sent as a response to a TuneReq message. It provides information about the currently selected DAB frequency and reception conditions.

The Tune command is used to select a specified DAB frequency. The tuneReq request initiates the Tune command. tuneCnf finalizes the Tune command and provides information about the reception state. This includes the selected frequency, the detected transmission mode and the synchronization state of the receiver.

See Also:

[tuneReq](#), [searchReq](#), [searchCnf](#)

searchCnf

```
public void searchCnf(dab.events.SearchCnfEvent e)
```

The searchCnf method finalizes a Search command and provides information about the command status, currently selected DAB frequency and current reception conditions.

The Search command is used in order to search for a DAB Ensemble according to a specified search mode. Searching for a DAB Ensemble can take a large amount of time. The start of searching is indicated by a 'Started' searchNtf message. Other searchNtf messages inform a DAB client about search progress. It is finalized by delivery of the searchCnf message. It informs about the command status, the selected frequency and the synchronization state. No further searchNtf messages will be delivered after the delivery of the searchCnf message.

See Also:

[searchReq](#), [searchNtf](#)

searchNtf

```
public void searchNtf(dab.events.SearchNtfEvent e)
```

A SearchNtf event is sent after a search for a DAB Ensemble was started searchReq. It informs about the start of searching and about the progress of searching. A SearchCnf event finalizes a Search command. No more SearchNtf events are sent after a SearchCnf event was sent.

The SearchNtf event is sent after the searching for a DAB Ensemble has been started and while searching is in progress in order to provide information about the current status of searching. The 'Started' notification is sent in any case. Progress notifications are only sent if notifications have been requested with the related SearchReq message. No further notifications will be sent after a SearchCnf message is delivered.

See Also:

[searchReq](#), [searchCnf](#)

scanCnf

```
public void scanCnf(dab.events.ScanCnfEvent e)
```

The ScanCnf message finalizes a Scan command. It informs about the result of scanning and the current tune state.

The Scan command is used in order to perform a search for all available DAB Ensembles in a specified frequency range. Depending on the frequency range and the search mode this operation may require a substantial amount of time from only a second up to several minutes. The command is started by the ScanReq message and is finished with the ScanCnf message. In between ScanNtf messages are sent in order to inform about the current status of searching if notifications are requested.

The ScanCnf message indicates that the scan command is finished and informs about the current tune state. As a result of performing the scan command the service information database is filled with information. If a SI subscription is running several SINtf messages are delivered to the connected application.

See Also:

[scanReq](#), [scanNtf](#)

scanNtf

```
public void scanNtf(dab.events.ScanNtfEvent e)
```

The ScanNtf message is sent after a search for all available DAB Ensembles in a specified frequency range is started by the ScanReq message. The ScanNtf message provides information about the current status of searching for all available DAB Ensembles in a specified frequency range. It is delivered to the connected application after the search has been started by the ScanReq message and notifications have been requested. No further notifications will be sent after a ScanCnf message is delivered.

See Also:

[scanReq](#), [scanCnf](#)

selectSICnf

```
public void selectSICnf(dab.events.SelectSICnfEvent e)
```

The selectSICnf method finalizes a SelectSI command and indicates current settings. The SelectSI command starts, stops or changes subscription to Service Directory Information.

The SelectSI command allows a DAB client to subscribe for Service Directory Information. The subscription is requested by the selectSIReg method and is confirmed with the SelectSICnf method. The subscription level can be changed by another SelectSI command. This includes the termination of subscription. After a successful subscription a connected client receives siNtf calls when the Service Directory changes.

See Also:

[selectSIReg](#), [siNtf](#)

siNtf

```
public void siNtf(dab.events.SINtfEvent e)
```

The siNtf notification is sent as a consequence of subscribing to Service Directory Information.

A call to siNtf indicates that the Service Directory has changed. The type of change is signaled and a handle to the changed service element is provided. If AutoDelivery is activated the changed information object itself is delivered together with the notification. Otherwise it can be requested with getEnsembleInfo, getServiceInfo or getComponentInfo. siNtf message is called as a result of the subscription to Service Directory Information.

See Also:

[selectSIReg](#), [selectSICnf](#)

getEnsembleInfoCnf

```
public void getEnsembleInfoCnf(dab.events.GetEnsembleInfoCnfEvent e)
```

The GetEnsembleInfoCnf method finalizes the GetEnsembleInfo command and delivers information about a requested DAB Ensemble to a DAB client.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

The GetEnsembleInfo command provides a DAB client with information about a specified DAB Ensemble, e.g. Label, No of Services, and so on. The command is initiated by a getEnsembleInfoReq request and is finished by a getEnsembleInfoCnf call.

See Also:

[getEnsembleInfoReq](#), [EnsembleInfo](#)

getServiceInfoCnf

```
public void getServiceInfoCnf(dab.events.GetServiceInfoCnfEvent e)
```

A call to the getServiceInfoCnf method finalizes the GetServiceInfo command and delivers information about a requested DAB Service to a DAB client.

The GetServiceInfo command provides a DAB client with information about a specified DAB Service, e.g. Label, No of Services, and so on. The command is initiated by a getServiceInfoReq message and is finished by a getServiceInfoCnf message.

See Also:

[getServiceInfoReq](#)

GetComponentInfoCnf

```
public void GetComponentInfoCnf(dab.events.GetComponentInfoCnfEvent e)
```

The GetComponentInfoCnf message finalizes the GetComponentInfo command and delivers information about a requested DAB Component to a DAB client.

The GetComponentInfo command provides a DAB client with information about a specified DAB Component, e.g. Label, Language and so on. The command is initiated by a GetComponentInfoReq request and is finished by a call to GetComponentInfoCnf message.

See Also:

[GetComponentInfoReq](#), [ComponentInfo](#)

selectReceptionInfoCnf

```
public void selectReceptionInfoCnf(dab.events.SelectReceptionInfoCnfEvent e)
```

The selectReceptionInfoCnf method finalizes the SelectReceptionInfo command. It informs about the command status and the current subscription level.

The selectReceptionInfo method allows a DAB client to subscribe for state change notifications concerning reception conditions in terms of synchronization, biterrorrate and audio decoder muting. The subscription is requested by selectReceptionInfoReq and is confirmed with selectReceptionInfoCnf. The subscription level can be changed by another SelectReceptionInfo command. This includes stopping of subscription. After a successful subscription the calling DAB client receives receptionInfoNtf calls when state changes occur.

See Also:

[selectReceptionInfoReq](#), [receptionInfoNtf](#)

receptionInfoNtf

```
public void receptionInfoNtf(dab.events.ReceptionInfoNtfEvent e)
```

The receptionInfoNtf method is called as a consequence of subscribing to state changes in synchronization, biterrorrate and audio decoder muting.

receptionInfoNtf indicates that the synchronization state, biterrorrate or mute state has changed (cf. ReceptionInfoNtfEvent). The ReceptionInfoNtf message is provided to a connected client as a result of subscription to state change notifications concerning reception conditions (selectReceptionInfoReq and selectReceptionInfoCnf messages).

See Also:

[selectReceptionInfoReq](#), [selectReceptionInfoCnf](#)

selectComponentCnf

```
public void selectComponentCnf(dab.events.SelectComponentCnfEvent e)
```

The SelectComponentCnf confirmation finalizes the SelectComponent command. It informs about the command status and the selection status of the specified component.

The SelectComponent command allows to start or stop applications delivered in DAB components. In general more than one component of the same DAB Ensemble can be selected simultaneously. It is possible to select one audio component, all programme-associated data components of the selected audio component and more than one independent data component at the same time. The selection of a component is requested by the selectComponentReq message and is confirmed by a selectComponentCnf call. It is possible that a component is removed from a DAB Ensemble which means it is no longer broadcast and therefore no longer available. This is indicated by a SINtf call and means that the selection is removed automatically.

See Also:

[selectComponentReq](#), [siNtf](#)

selectComponentStreamCnf

```
public void selectComponentStreamCnf(dab.events.SelectComponentStreamCnfEvent e)
```

The `selectComponentStreamCnf` method returns the requested stream (if the command was succesful) and informs about the result of the command.

See Also:

[selectComponentStreamReq](#)

componentNtf

```
public void componentNtf(dab.events.ComponentNtfEvent e)
```

The `componentNtf` method is called if there are changes to the selection mode of a component. This typically happens, when the selection of a component is stopped.

Note, that this notification will be produced due to internal reasons (e.g. after tuning to another ensemble) and not in response to a `selectComponentReq` request (that is handled by `selectComponentCnf`).

selectObjectCnf

```
public void selectObjectCnf(dab.events.SelectObjectCnfEvent e)
```

The `SelectObjectCnf` method finalizes the `SelectObject` command. The `SelectObject` command selects an object from a selected DAB Component. This includes requesting an object from a data component, delivery after reception and notification of updates as long as the object is selected.

The `SelectObject` command selects an object from a selected component. Selection means it is requested for delivery and if wanted also updates of the object are delivered. Additionally it is possible to give some hints for caching. More than one object and also from more than one component can be selected simultanously. The selection of an object is requested by `selectObjectReq` and is confirmed by calling `selectObjectCnf`. The object is delivered using `objectNtf`. This includes first-time delivery and all updates. Beyond starting or stopping a selection it is possible to remove all other selections belonging to the same component by setting parameter `replaceSelections` to true. It is possible that a component is removed from a DAB Ensemble. This is indicated by a call to `siNtf`. In this case also the selected objects of the service are no longer selected. It is possible that an object is removed from current on-air service. This is indicated by calling `objectNtf`. In this case the selections for this object are automatically disabled. Currently object selection makes only sense with applications of type `BroadcastWebSite`. Objects of applications like Slideshows or Dynamic Label are delivered automatically using `objectNtf`.

See Also:

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

[selectObjectReq](#), [objectNtf](#)

selectApplicationCnf

```
public void selectApplicationCnf(dab.events.SelectApplicationCnfEvent e)
```

The method is called as a consequence of selecting an application from a data component by use of the `SelectApplication` command. It delivers a proxy to control the application.

See Also:

[selectApplicationReq](#)

objectNtf

```
public void objectNtf(dab.events.ObjectNtfEvent e)
```

The `objectNtf` method is called as a consequence of selecting objects from a data component by use of the `SelectObject` command. It delivers a selected object partially or complete to a DAB client.

`objectNtf` is used to deliver a selected object to the connected DAB client. Depending on the request mode the object is delivered only once or more than once in case of updates. If the object can not be delivered in-time as indicated by a call to `selectObjectCnf`, then `objectNtf` informs about the delay. If transmission of an selected object is stopped, `objectNtf` informs about the termination of the object transmission and the object selection. It is possible that a DAB Component is removed from a DAB Ensemble. This is indicated by a call to `siNtf`. In this case also the selected objects of the component are no longer selected. No termination messages are sent for terminated object selections resulting from termination of a component.

See Also:

[selectObjectReq](#), [selectObjectCnf](#), [DABObject](#)

getLocationInfoCnf

```
public void getLocationInfoCnf(dab.events.GetLocationInfoCnfEvent e)
```

`getLocationInfoCnf` confirms the `getLocationInfo` command. This means the delivery of location information will start from now on.

See Also:

[getLocationInfoReq](#), [locationInfoNtf](#)

locationInfoNtf

```
public void locationInfoNtf(dab.events.LocationInfoNtfEvent e)
```

`locationInfoNtf` notifies about location information.

See Also:

[getLocationInfoReq](#), [getLocationInfoCnf](#)

conflictResolutionNtf

```
public void conflictResolutionNtf(dab.events.ConflictResolutionNtfEvent e)
```

The method is called for notifying the listener of resource conflicts. The listener can react to this event using the request `respondConflictResolutionReq`.

See Also:

[respondConflictResolutionReq](#), [respondConflictResolutionCnf](#)

respondConflictResolutionCnf

```
public void respondConflictResolutionCnf(dab.events.RespondConflictResolutionCnfEvent e)
```

The method is called for confirming a reaction to a resource conflict.

See Also:

[respondConflictResolutionReq](#), [conflictResolutionNtf](#)

operationControlCnf

```
public void operationControlCnf(dab.events.OperationControlCnfEvent e)
```

The confirmation indicates the result of the `operationControl` command.

See Also:

[operationControlReq](#)

serviceFollowingNtf

```
public void serviceFollowingNtf(dab.events.ServiceFollowingNtfEvent e)
```

The notification informs about service following actions.

See Also:

[operationControlReq](#)

drcModeNtf

```
public void drcModeNtf(dab.events.DRCModeNtfEvent e)
```

The notification informs about DRC mode changes.

See Also:

[operationControlReq](#)

systemFailureNtf

```
public void systemFailureNtf(dab.events.SystemFailureNtfEvent e)
```

SystemFailureNtf notifies about severe problems with the hardware or the middleware (e.g. breakdown of the communication to the DAB receiver). This should not be confused with the indication of errors for a particular command, which relates only to the command itself.

Typically, after the notification is sent, the package can no longer be used or needs to be reinitialised.

Class DABException

[dab](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

```
java.lang.Object
|
+-- java.lang.Throwable
    |
    +-- java.lang.Exception
        |
        +-- dab.DABException
```

All Implemented Interfaces:

java.io.Serializable

Direct Known Subclasses:

[DABConnectionException](#), [DABInvalidURLException](#), [DABNotAvailableException](#), [ResourceConflictException](#)

```
public Class DABException
```

```
extends java.lang.Exception
```

DABException is the superclass for exceptions inside the DAB package.

Version:

1.0

Constructor Summary	Page
DABException ()	286
DABException (java.lang.String msg)	286

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail**DABException**

```
public DABException()
```

DABException

```
public DABException(java.lang.String msg)
```

Class DABConstants

[dab](#)

```
java.lang.Object
```

```
|
```

```
+--dab.DABConstants
```

```
public Class DABConstants
```

```
extends java.lang.Object
```

DABConstants contains the constants that are used inside the whole package (including the subpackages).

If you would like to add any new constants, please contact WorldDAB Information and Registration Centre, Wyvil Court, Wyvil Road, LONDON SW8 2TG, England, Tel: +44 171 896 90 51, Fax: +44 171 896 90 55, E-mail: worlddab-irc@worlddab.org.

Version:

1.07

Field Summary		Page
static final int	acsEuroCryptEN50094	316
static final int	acsNone	316
static final int	acsNR_MSK	316
static final int	acsReserverd1	316
static final int	acsReserverd2	316
static final int	acsReserverd3	316
static final int	acsReserverd4	316
static final int	acsReserverd5	316
static final int	announcementAlarm	309
static final int	announcementAreaWeatherFlash	310
static final int	announcementEventAnnouncement	310
static final int	announcementNewsFlash	310
static final int	announcementReserved1	310
static final int	announcementReserved2	310
static final int	announcementReserved3	310
static final int	announcementReserved4	310
static final int	announcementReserved5	310
static final int	announcementReserved6	310
static final int	announcementReserved7	310
static final int	announcementReserved8	310
static final int	announcementRoadTrafficFlash	309
static final int	announcementSpecialEvent	310
static final int	announcementTransportFlash	309
static final int	announcementWarning_Service	309
static final int	bitErrorRateLevel1	307
static final int	bitErrorRateLevel2	307
static final int	bitErrorRateLevel3	307
static final int	bitErrorRateLevel4	307
static final int	bitErrorRateLevel5	307
static final int	bitErrorRateLevelUnknown	307
static final int	charsetCompleteEBULatin	328
static final int	charsetEBUArabic_HebrewETC	328
static final int	charsetEBUCyrillicGreek	328
static final int	charsetISOLatinAlphabetNo2	328

static final int	<u>componentTypeBackgroundSound</u>	317
static final int	<u>componentTypeBroadcastWebSite</u>	317
static final int	<u>componentTypeDynamicLabel</u>	317
static final int	<u>componentTypeEmergencyWarningSystem</u>	317
static final int	<u>componentTypeForegroundSound</u>	316
static final int	<u>componentTypeInteractiveTextTransmissionSystem</u>	317
static final int	<u>componentTypeIPTunneling</u>	317
static final int	<u>componentTypeJava</u>	317
static final int	<u>componentTypeMultichannelAudio</u>	317
static final int	<u>componentTypePaging</u>	317
static final int	<u>componentTypeSlideshow</u>	317
static final int	<u>componentTypeTrafficMessageChannel</u>	317
static final int	<u>componentTypeUnspecified</u>	316
static final int	<u>conflictResolutionAnswerNo</u>	333
static final int	<u>conflictResolutionAnswerYes</u>	333
static final int	<u>conflictResolutionOperationGetComponentInfoReq</u>	332
static final int	<u>conflictResolutionOperationGetEnsembleInfoReq</u>	332
static final int	<u>conflictResolutionOperationGetLocationInfoReq</u>	332
static final int	<u>conflictResolutionOperationGetServiceInfoReq</u>	332
static final int	<u>conflictResolutionOperationNone</u>	331
static final int	<u>conflictResolutionOperationOperationControlReq</u>	332
static final int	<u>conflictResolutionOperationScanReq</u>	332
static final int	<u>conflictResolutionOperationSearchReq</u>	331
static final int	<u>conflictResolutionOperationSelectApplicationReq</u>	332
static final int	<u>conflictResolutionOperationSelectComponentReq</u>	332
static final int	<u>conflictResolutionOperationSelectComponentStreamReq</u>	332
static final int	<u>conflictResolutionOperationSelectObjectReq</u>	332
static final int	<u>conflictResolutionOperationSelectReceptionInfoReq</u>	332
static final int	<u>conflictResolutionOperationSelectSIReq</u>	332
static final int	<u>conflictResolutionOperationTuneReq</u>	331
static final int	<u>conflictResolutionSuboperationNone</u>	333
static final int	<u>conflictResolutionTurnPreempt</u>	331
static final int	<u>conflictResolutionTurnProbe</u>	331
static final int	<u>conflictResolutionTurnProceed</u>	331

static final int	<u>conflictResolutionTurnStop</u>	331
static final int	<u>countryAlbania</u>	311
static final int	<u>countryAlgeria</u>	311
static final int	<u>countryAndorra</u>	311
static final int	<u>countryAustria</u>	311
static final int	<u>countryAzores_Portugal</u>	311
static final int	<u>countryBelarus</u>	311
static final int	<u>countryBelgium</u>	311
static final int	<u>countryBosniaHerzegovina</u>	311
static final int	<u>countryBulgaria</u>	311
static final int	<u>countryCanaries_Spain</u>	311
static final int	<u>countryCroatia</u>	311
static final int	<u>countryCyprus</u>	311
static final int	<u>countryCzechRepublic</u>	312
static final int	<u>countryDenmark</u>	312
static final int	<u>countryEgypt</u>	312
static final int	<u>countryEstonia</u>	312
static final int	<u>countryFaroe_Denmark</u>	312
static final int	<u>countryFinland</u>	312
static final int	<u>countryFrance</u>	312
static final int	<u>countryGermany1</u>	312
static final int	<u>countryGermany2</u>	312
static final int	<u>countryGibraltar_UnitedKingdom</u>	312
static final int	<u>countryGreece</u>	312
static final int	<u>countryHungary</u>	312
static final int	<u>countryIceland</u>	313
static final int	<u>countryIraq</u>	313
static final int	<u>countryIreland</u>	313
static final int	<u>countryIsrael</u>	313
static final int	<u>countryItaly</u>	313
static final int	<u>countryJordan</u>	313
static final int	<u>countryLatvia</u>	313
static final int	<u>countryLebanon</u>	313
static final int	<u>countryLibya</u>	313

static final int	<u>countryLiechtenstein</u>	313
static final int	<u>countryLithuania</u>	313
static final int	<u>countryLuxembourg</u>	313
static final int	<u>countryMacedonia</u>	314
static final int	<u>countryMadeira_Portugal</u>	314
static final int	<u>countryMalta</u>	314
static final int	<u>countryMoldova</u>	314
static final int	<u>countryMonaco</u>	314
static final int	<u>countryMorocco</u>	314
static final int	<u>countryNetherlands</u>	314
static final int	<u>countryNorways</u>	314
static final int	<u>countryPalestine</u>	314
static final int	<u>countryPoland</u>	314
static final int	<u>countryPortugal</u>	314
static final int	<u>countryRomania</u>	314
static final int	<u>countryRussianFederation</u>	315
static final int	<u>countrySanMarino</u>	315
static final int	<u>countrySlovakia</u>	315
static final int	<u>countrySlovenia</u>	315
static final int	<u>countrySpain</u>	315
static final int	<u>countrySweden</u>	315
static final int	<u>countrySwitzerland</u>	315
static final int	<u>countrySyrianArabRepublic</u>	315
static final int	<u>countryTunisia</u>	315
static final int	<u>countryTurkey</u>	315
static final int	<u>countryUkraine</u>	315
static final int	<u>countryUnitedKingdom</u>	315
static final int	<u>countryVaticanCityState</u>	316
static final int	<u>countryYugoslavia</u>	316
static final int	<u>deliveryModeComplete</u>	304
static final int	<u>deliveryModePartial</u>	304
static final int	<u>language2C</u>	321
static final int	<u>language2D</u>	321
static final int	<u>language2E</u>	321

static final int	<u>language2F</u>	321
static final int	<u>language41</u>	328
static final int	<u>language42</u>	328
static final int	<u>language43</u>	328
static final int	<u>language44</u>	328
static final int	<u>languageAlbanian</u>	318
static final int	<u>languageAmharic</u>	323
static final int	<u>languageArabic</u>	323
static final int	<u>languageArmenian</u>	323
static final int	<u>languageAssamese</u>	323
static final int	<u>languageAzerbaijani</u>	323
static final int	<u>languageBackgroundSoundCleanFeed</u>	328
static final int	<u>languageBambora</u>	323
static final int	<u>languageBasque</u>	319
static final int	<u>languageBelorussian</u>	323
static final int	<u>languageBengali</u>	323
static final int	<u>languageBreton</u>	318
static final int	<u>languageBulgarian</u>	323
static final int	<u>languageBurmese</u>	324
static final int	<u>languageCatalan</u>	318
static final int	<u>languageChinese</u>	324
static final int	<u>languageChurash</u>	324
static final int	<u>languageCroatian</u>	318
static final int	<u>languageCzech</u>	318
static final int	<u>languageDanish</u>	318
static final int	<u>languageDari</u>	324
static final int	<u>languageDutch</u>	320
static final int	<u>languageEnglish</u>	318
static final int	<u>languageEsperanto</u>	318
static final int	<u>languageEstonian</u>	318
static final int	<u>languageFaroese</u>	319
static final int	<u>languageFinnish</u>	321
static final int	<u>languageFlemish</u>	321
static final int	<u>languageFrench</u>	319

static final int	<u>languageFrisian</u>	319
static final int	<u>languageFulani</u>	324
static final int	<u>languageGaelic</u>	319
static final int	<u>languageGalician</u>	319
static final int	<u>languageGeorgian</u>	324
static final int	<u>languageGerman</u>	318
static final int	<u>languageGreek</u>	324
static final int	<u>languageGujurati</u>	324
static final int	<u>languageGurani</u>	324
static final int	<u>languageHausa</u>	324
static final int	<u>languageHebrew</u>	324
static final int	<u>languageHindi</u>	324
static final int	<u>languageHungarian</u>	320
static final int	<u>languageIcelandic</u>	319
static final int	<u>languageIndonesian</u>	325
static final int	<u>languageIrish</u>	319
static final int	<u>languageItalian</u>	319
static final int	<u>languageJapanese</u>	325
static final int	<u>languageKannada</u>	325
static final int	<u>languageKazakh</u>	325
static final int	<u>languageKhmer</u>	325
static final int	<u>languageKorean</u>	325
static final int	<u>languageLaotian</u>	325
static final int	<u>languageLappish</u>	319
static final int	<u>languageLatin</u>	319
static final int	<u>languageLatvian</u>	319
static final int	<u>languageLithuanian</u>	320
static final int	<u>languageLuxembourgian</u>	320
static final int	<u>languageMacedonian</u>	325
static final int	<u>languageMalagasay</u>	325
static final int	<u>languageMalaysian</u>	325
static final int	<u>languageMaltese</u>	320
static final int	<u>languageMarathi</u>	325
static final int	<u>languageMoldavian</u>	325

static final int	<u>languageNational30</u>	321
static final int	<u>languageNational31</u>	322
static final int	<u>languageNational32</u>	322
static final int	<u>languageNational33</u>	322
static final int	<u>languageNational34</u>	322
static final int	<u>languageNational35</u>	322
static final int	<u>languageNational36</u>	322
static final int	<u>languageNational37</u>	322
static final int	<u>languageNational38</u>	322
static final int	<u>languageNational39</u>	322
static final int	<u>languageNational3A</u>	322
static final int	<u>languageNational3B</u>	322
static final int	<u>languageNational3C</u>	322
static final int	<u>languageNational3D</u>	323
static final int	<u>languageNational3E</u>	323
static final int	<u>languageNational3F</u>	323
static final int	<u>languageNdebele</u>	326
static final int	<u>languageNepali</u>	326
static final int	<u>languageNorwegian</u>	320
static final int	<u>languageOccitan</u>	320
static final int	<u>languageOriya</u>	326
static final int	<u>languagePapamiento</u>	326
static final int	<u>languagePersian</u>	326
static final int	<u>languagePolish</u>	320
static final int	<u>languagePortuguese</u>	320
static final int	<u>languagePunjabi</u>	326
static final int	<u>languagePushtu</u>	326
static final int	<u>languageQuechua</u>	326
static final int	<u>languageRomanian</u>	320
static final int	<u>languageRomansh</u>	320
static final int	<u>languageRussian</u>	326
static final int	<u>languageRuthenian</u>	326
static final int	<u>languageSerbian</u>	320
static final int	<u>languageSerbo_Croat</u>	326

static final int	<u>languageShona</u>	326
static final int	<u>languageSinhalese</u>	327
static final int	<u>languageSlovak</u>	321
static final int	<u>languageSlovene</u>	321
static final int	<u>languageSomali</u>	327
static final int	<u>languageSpanish</u>	318
static final int	<u>languageSranan_Tongo</u>	327
static final int	<u>languageSwahili</u>	327
static final int	<u>languageSwedish</u>	321
static final int	<u>languageTadzhik</u>	327
static final int	<u>languageTamil</u>	327
static final int	<u>languageTatar</u>	327
static final int	<u>languageTelugu</u>	327
static final int	<u>languageThai</u>	327
static final int	<u>languageTurkish</u>	321
static final int	<u>languageUkrainian</u>	327
static final int	<u>languageUnkown</u>	317
static final int	<u>languageUrdu</u>	327
static final int	<u>languageUzbek</u>	327
static final int	<u>languageVietnamese</u>	328
static final int	<u>languageWalloon</u>	321
static final int	<u>languageWelsh</u>	318
static final int	<u>languageZulu</u>	328
static final int	<u>locationInfoOnce</u>	329
static final int	<u>locationInfoPeriodByDistance</u>	329
static final int	<u>locationInfoPeriodByTime</u>	329
static final int	<u>locationInfoPosition</u>	329
static final int	<u>locationInfoRegionId</u>	329
static final int	<u>locationInfoStop</u>	329
static final int	<u>muteStateMuting</u>	307
static final int	<u>muteStateNotMuting</u>	307
static final int	<u>muteStatePartialMuting</u>	307
static final int	<u>muteStateUnknown</u>	307
static final int	<u>notification16kHzStep</u>	302

static final int	<u>notificationComponentAdded</u>	303
static final int	<u>notificationComponentChanged</u>	303
static final int	<u>notificationComponentRemoved</u>	303
static final int	<u>notificationEnsembleAdded</u>	302
static final int	<u>notificationEnsembleChanged</u>	303
static final int	<u>notificationEnsembleFound</u>	302
static final int	<u>notificationEnsembleRemoved</u>	302
static final int	<u>notificationFrequencyStep</u>	302
static final int	<u>notificationNone</u>	302
static final int	<u>notificationOff</u>	302
static final int	<u>notificationSearchStarted</u>	302
static final int	<u>notificationServiceAdded</u>	303
static final int	<u>notificationServiceChanged</u>	303
static final int	<u>notificationServiceRemoved</u>	303
static final int	<u>notificationTableEntry</u>	302
static final int	<u>operationControlGetDRCMode</u>	330
static final int	<u>operationControlGetDRCModeNotifications</u>	330
static final int	<u>operationControlGetServiceFollowing</u>	330
static final int	<u>operationControlGetServiceFollowingNotifications</u>	330
static final int	<u>operationControlGetVolume</u>	329
static final int	<u>operationControlSetDRCMode</u>	330
static final int	<u>operationControlSetServiceFollowing</u>	329
static final int	<u>operationControlSetVolume</u>	329
static final int	<u>requestModeOff</u>	303
static final int	<u>requestModeOnce</u>	304
static final int	<u>requestModeUpdate</u>	304
static final int	<u>resultApplicationNotFound</u> the requested/indicated application was not found	300
static final int	<u>resultComponentNotFound</u> the requested/indicated component was not found	300
static final int	<u>resultEnsembleNotFound</u> the requested/indicated ensemble was not found	300
static final int	<u>resultFatalError</u> a system error occurred (either related to hardware or to the operating system)	299

static final int	<u>resultInternalError</u> an internal error occurred in the DAB VM (e.g. an implementation error)	299
static final int	<u>resultInvalidParameter</u> the value of some parameter is not correct	299
static final int	<u>resultNonApplicableFunction</u> the operation is not applicable in the current context	300
static final int	<u>resultNotSupported</u> the requested operation is not supported	299
static final int	<u>resultObjectNotSelected</u> the indicated object was not be selected (in advance)	300
static final int	<u>resultOK</u> no problems occurred	299
static final int	<u>resultOutOfMemory</u> the system ran out of memory	300
static final int	<u>resultServiceNotFound</u> the requested/indicated service was not found	300
static final int	<u>searchCanadaFrequencyTableLBand</u>	302
static final int	<u>searchCEPTFrequencyTableBandIII</u>	302
static final int	<u>searchCEPTFrequencyTableLBand</u>	302
static final int	<u>searchModel6kHzSteps</u>	301
static final int	<u>searchModeAutomatic</u>	301
static final int	<u>searchModeContinuous</u>	301
static final int	<u>searchModeDown</u>	301
static final int	<u>searchModeUp</u>	301
static final int	<u>searchModeUseFrequencyRange</u>	301
static final int	<u>searchModeUseTables</u>	301
static final int	<u>selectionModeAdd</u>	303
static final int	<u>selectionModeRemove</u>	303
static final int	<u>selectionModeRemoveAll</u>	303
static final int	<u>selectionModeReplace</u>	303
static final int	<u>selectionStateDelayed</u>	308
static final int	<u>selectionStateOk</u>	307
static final int	<u>selectionStateTerminated</u>	308
static final int	<u>serviceElementTypeComponent</u>	329
static final int	<u>serviceElementTypeEnsemble</u>	329

static final int	<u>serviceElementTypeService</u>	329
static final int	<u>serviceElementTypeUndefined</u>	328
static final int	<u>serviceFollowingLeavingService</u>	330
static final int	<u>serviceFollowingSelectingService</u>	330
static final int	<u>serviceFollowingTryingAlternativeService</u>	330
static final int	<u>serviceSelectorAnnouncement</u>	309
static final int	<u>serviceSelectorAnnouncementSupport</u>	309
static final int	<u>serviceSelectorCountry</u>	308
static final int	<u>serviceSelectorDate</u>	308
static final int	<u>serviceSelectorDynamicProgrammeType</u>	308
static final int	<u>serviceSelectorFrequency</u>	308
static final int	<u>serviceSelectorLabel</u>	308
static final int	<u>serviceSelectorLanguage</u>	309
static final int	<u>serviceSelectorNone</u>	308
static final int	<u>serviceSelectorRegion</u>	308
static final int	<u>serviceSelectorRegionId</u>	309
static final int	<u>serviceSelectorRegionLabel</u>	309
static final int	<u>serviceSelectorStartObject</u>	309
static final int	<u>serviceSelectorStaticProgrammeType</u>	308
static final int	<u>serviceSelectorTime</u>	308
static final int	<u>serviceSelectorTimeOffset</u>	308
static final int	<u>serviceTypeAudioService</u>	309
static final int	<u>serviceTypeDataService</u>	309
static final int	<u>streamTypeAudio</u>	330
static final int	<u>streamTypeFIDC</u>	331
static final int	<u>streamTypePacket</u>	330
static final int	<u>streamTypeStream</u>	330
static final int	<u>streamTypeXPAD</u>	330
static final int	<u>subscriberInfoExpiredSubscription</u>	331
static final int	<u>subscriberInfoNoAlgorithm</u>	331
static final int	<u>subscriberInfoNoCA</u>	331
static final int	<u>subscriberInfoNoSubscription</u>	331
static final int	<u>syncStateDABSignalDetected</u>	304
static final int	<u>syncStateFICReadable</u>	304

static final int	<u>syncStateNotSynchronized</u>	304
static final int	<u>syncStateSynchronizationStateUnknown</u>	304
static final int	<u>syncStateTimeAndFrequencySynchronized</u>	304
static final int	<u>syncUpdateBitErrorRateState</u>	306
static final int	<u>syncUpdateMuteState</u>	307
static final int	<u>syncUpdateSynchronizationState</u>	306
static final int	<u>transmissionModel1</u>	301
static final int	<u>transmissionMode2</u>	301
static final int	<u>transmissionMode3</u>	301
static final int	<u>transmissionMode4</u>	301
static final int	<u>transmissionModeAutomatic</u>	300
static final int	<u>transmissionModeUnknown</u>	301
static final int	<u>tuneStateNotTuned</u>	304
static final int	<u>tuneStateSearching</u>	305
static final int	<u>tuneStateTuned</u>	305
static final int	<u>tuneStateTuning</u>	304
static final int	<u>tuneStateUnknown</u>	304
static final int	<u>updatedAnnouncement</u>	306
static final int	<u>updatedAnnouncementSupport</u>	306
static final int	<u>updatedAudioComponent</u>	306
static final int	<u>updatedBitrate</u>	306
static final int	<u>updatedCountry</u>	305
static final int	<u>updatedDate</u>	305
static final int	<u>updatedDynamicProgrammeType</u>	305
static final int	<u>updatedFrequency</u>	305
static final int	<u>updatedLabel</u>	305
static final int	<u>updatedLanguage</u>	306
static final int	<u>updatedNone</u>	305
static final int	<u>updatedObjectDirectory</u>	306
static final int	<u>updatedProgrammeNumber</u>	306
static final int	<u>updatedRegion</u>	305
static final int	<u>updatedRegionId</u>	306
static final int	<u>updatedRegionLabel</u>	306
static final int	<u>updatedStartObject</u>	306

static final int	updatedStaticProgrammeType	305
static final int	updatedTime	305
static final int	updatedTimeOffset	305

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Summary	<i>Page</i>
DABConstants ()	333

Method Summary	<i>Page</i>	
static java.lang.String	result2String (int result) The method returns a string for the given result code.	333

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

resultOK

public static final int **resultOK**

no problems occurred

resultNotSupported

public static final int **resultNotSupported**

the requested operation is not supported

resultFatalError

public static final int **resultFatalError**

a system error occurred (either related to hardware or to the operating system)

resultInternalError

public static final int **resultInternalError**

an internal error occurred in the DAB VM (e.g. an implementation error)

resultInvalidParameter

public static final int **resultInvalidParameter**

the value of some parameter is not correct

resultOutOfMemory

```
public static final int resultOutOfMemory
```

the system ran out of memory

resultNonApplicableFunction

```
public static final int resultNonApplicableFunction
```

the operation is not applicable in the current context

resultEnsembleNotFound

```
public static final int resultEnsembleNotFound
```

the requested/indicated ensemble was not found

resultServiceNotFound

```
public static final int resultServiceNotFound
```

the requested/indicated service was not found

resultComponentNotFound

```
public static final int resultComponentNotFound
```

the requested/indicated component was not found

resultObjectNotSelected

```
public static final int resultObjectNotSelected
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

the indicated object was not be selected (in advance)

resultApplicationNotFound

```
public static final int resultApplicationNotFound
```

the requested/indicated application was not found

transmissionModeAutomatic

```
public static final int transmissionModeAutomatic
```

transmissionMode1

```
public static final int transmissionMode1
```

transmissionMode2

```
public static final int transmissionMode2
```

transmissionMode3

```
public static final int transmissionMode3
```

transmissionMode4

```
public static final int transmissionMode4
```

transmissionModeUnknown

```
public static final int transmissionModeUnknown
```

searchModeAutomatic

```
public static final int searchModeAutomatic
```

searchMode16kHzSteps

```
public static final int searchMode16kHzSteps
```

searchModeUp

```
public static final int searchModeUp
```

searchModeDown

```
public static final int searchModeDown
```

searchModeUseTables

```
public static final int searchModeUseTables
```

searchModeUseFrequencyRange

```
public static final int searchModeUseFrequencyRange
```

searchModeContinuous

```
public static final int searchModeContinuous
```

searchCEPTFrequencyTableBandIII

```
public static final int searchCEPTFrequencyTableBandIII
```

searchCEPTFrequencyTableLBand

```
public static final int searchCEPTFrequencyTableLBand
```

searchCanadaFrequencyTableLBand

```
public static final int searchCanadaFrequencyTableLBand
```

notificationOff

```
public static final int notificationOff
```

notificationFrequencyStep

```
public static final int notificationFrequencyStep
```

notification16kHzStep

```
public static final int notification16kHzStep
```

notificationTableEntry

```
public static final int notificationTableEntry
```

notificationEnsembleFound

```
public static final int notificationEnsembleFound
```

notificationSearchStarted

```
public static final int notificationSearchStarted
```

notificationNone

```
public static final int notificationNone
```

notificationEnsembleAdded

```
public static final int notificationEnsembleAdded
```

notificationEnsembleRemoved

```
public static final int notificationEnsembleRemoved
```

notificationEnsembleChanged

```
public static final int notificationEnsembleChanged
```

notificationServiceAdded

```
public static final int notificationServiceAdded
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

notificationServiceRemoved

```
public static final int notificationServiceRemoved
```

notificationServiceChanged

```
public static final int notificationServiceChanged
```

notificationComponentAdded

```
public static final int notificationComponentAdded
```

notificationComponentRemoved

```
public static final int notificationComponentRemoved
```

notificationComponentChanged

```
public static final int notificationComponentChanged
```

selectionModeReplace

```
public static final int selectionModeReplace
```

selectionModeAdd

```
public static final int selectionModeAdd
```

selectionModeRemove

```
public static final int selectionModeRemove
```

selectionModeRemoveAll

```
public static final int selectionModeRemoveAll
```

requestModeOff

```
public static final int requestModeOff
```

requestModeOnce

```
public static final int requestModeOnce
```

requestModeUpdate

```
public static final int requestModeUpdate
```

deliveryModeComplete

```
public static final int deliveryModeComplete
```

deliveryModePartial

```
public static final int deliveryModePartial
```

syncStateSynchronizationStateUnknown

```
public static final int syncStateSynchronizationStateUnknown
```

syncStateNotSynchronized

```
public static final int syncStateNotSynchronized
```

syncStateDABSignalDetected

```
public static final int syncStateDABSignalDetected
```

syncStateTimeAndFrequencySynchronized

```
public static final int syncStateTimeAndFrequencySynchronized
```

syncStateFICReadable

```
public static final int syncStateFICReadable
```

tuneStateUnknown

```
public static final int tuneStateUnknown
```

tuneStateNotTuned

```
public static final int tuneStateNotTuned
```

tuneStateTuning

```
public static final int tuneStateTuning
```

tuneStateSearching

```
public static final int tuneStateSearching
```

tuneStateTuned

```
public static final int tuneStateTuned
```

updatedNone

```
public static final int updatedNone
```

updatedLabel

```
public static final int updatedLabel
```

updatedCountry

```
public static final int updatedCountry
```

updatedFrequency

```
public static final int updatedFrequency
```

updatedDate

```
public static final int updatedDate
```

updatedTime

```
public static final int updatedTime
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

updatedTimeOffset

```
public static final int updatedTimeOffset
```

updatedRegion

```
public static final int updatedRegion
```

updatedStaticProgrammeType

```
public static final int updatedStaticProgrammeType
```

updatedDynamicProgrammeType

```
public static final int updatedDynamicProgrammeType
```

updatedAnnouncement

```
public static final int updatedAnnouncement
```

updatedLanguage

```
public static final int updatedLanguage
```

updatedRegionId

```
public static final int updatedRegionId
```

updatedRegionLabel

```
public static final int updatedRegionLabel
```

updatedAnnouncementSupport

```
public static final int updatedAnnouncementSupport
```

updatedStartObject

```
public static final int updatedStartObject
```

updatedObjectDirectory

```
public static final int updatedObjectDirectory
```

updatedProgrammeNumber

```
public static final int updatedProgrammeNumber
```

updatedAudioComponent

```
public static final int updatedAudioComponent
```

updatedBitrate

```
public static final int updatedBitrate
```

syncUpdateSynchronizationState

```
public static final int syncUpdateSynchronizationState
```

syncUpdateBitErrorRateState

```
public static final int syncUpdateBitErrorRateState
```

syncUpdateMuteState

```
public static final int syncUpdateMuteState
```

bitErrorRateLevelUnknown

```
public static final int bitErrorRateLevelUnknown
```

bitErrorRateLevel1

```
public static final int bitErrorRateLevel1
```

bitErrorRateLevel2

```
public static final int bitErrorRateLevel2
```

bitErrorRateLevel3

```
public static final int bitErrorRateLevel3
```

bitErrorRateLevel4

```
public static final int bitErrorRateLevel4
```

bitErrorRateLevel5

```
public static final int bitErrorRateLevel5
```

muteStateUnknown

```
public static final int muteStateUnknown
```

muteStateMuting

```
public static final int muteStateMuting
```

muteStatePartialMuting

```
public static final int muteStatePartialMuting
```

muteStateNotMuting

```
public static final int muteStateNotMuting
```

selectionStateOk

```
public static final int selectionStateOk
```

selectionStateDelayed

```
public static final int selectionStateDelayed
```

selectionStateTerminated

```
public static final int selectionStateTerminated
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

serviceSelectorNone

```
public static final int serviceSelectorNone
```

serviceSelectorLabel

```
public static final int serviceSelectorLabel
```

serviceSelectorCountry

```
public static final int serviceSelectorCountry
```

serviceSelectorFrequency

```
public static final int serviceSelectorFrequency
```

serviceSelectorDate

```
public static final int serviceSelectorDate
```

serviceSelectorTime

```
public static final int serviceSelectorTime
```

serviceSelectorTimeOffset

```
public static final int serviceSelectorTimeOffset
```

serviceSelectorRegion

```
public static final int serviceSelectorRegion
```

serviceSelectorStaticProgrammeType

```
public static final int serviceSelectorStaticProgrammeType
```

serviceSelectorDynamicProgrammeType

```
public static final int serviceSelectorDynamicProgrammeType
```

serviceSelectorAnnouncement

```
public static final int serviceSelectorAnnouncement
```

serviceSelectorLanguage

```
public static final int serviceSelectorLanguage
```

serviceSelectorRegionId

```
public static final int serviceSelectorRegionId
```

serviceSelectorRegionLabel

```
public static final int serviceSelectorRegionLabel
```

serviceSelectorAnnouncementSupport

```
public static final int serviceSelectorAnnouncementSupport
```

serviceSelectorStartObject

```
public static final int serviceSelectorStartObject
```

serviceTypeAudioService

```
public static final int serviceTypeAudioService
```

serviceTypeDataService

```
public static final int serviceTypeDataService
```

announcementAlarm

```
public static final int announcementAlarm
```

announcementRoadTrafficFlash

```
public static final int announcementRoadTrafficFlash
```

announcementTransportFlash

```
public static final int announcementTransportFlash
```

announcementWarning_Service

```
public static final int announcementWarning_Service
```

announcementNewsFlash

```
public static final int announcementNewsFlash
```

announcementAreaWeatherFlash

```
public static final int announcementAreaWeatherFlash
```

announcementEventAnnouncement

```
public static final int announcementEventAnnouncement
```

announcementSpecialEvent

```
public static final int announcementSpecialEvent
```

announcementReserved1

```
public static final int announcementReserved1
```

announcementReserved2

```
public static final int announcementReserved2
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

announcementReserved3

```
public static final int announcementReserved3
```

announcementReserved4

```
public static final int announcementReserved4
```

announcementReserved5

```
public static final int announcementReserved5
```

announcementReserved6

```
public static final int announcementReserved6
```

announcementReserved7

```
public static final int announcementReserved7
```

announcementReserved8

```
public static final int announcementReserved8
```

countryAlbania

```
public static final int countryAlbania
```

countryAlgeria

```
public static final int countryAlgeria
```

countryAndorra

```
public static final int countryAndorra
```

countryAustria

```
public static final int countryAustria
```

countryAzores_Portugal

```
public static final int countryAzores_Portugal
```

countryBelgium

```
public static final int countryBelgium
```

countryBelarus

```
public static final int countryBelarus
```

countryBosniaHerzegovina

```
public static final int countryBosniaHerzegovina
```

countryBulgaria

```
public static final int countryBulgaria
```

countryCanaries_Spain

```
public static final int countryCanaries_Spain
```

countryCroatia

```
public static final int countryCroatia
```

countryCyprus

```
public static final int countryCyprus
```

countryCzechRepublic

```
public static final int countryCzechRepublic
```

countryDenmark

```
public static final int countryDenmark
```

countryEgypt

```
public static final int countryEgypt
```

countryEstonia

```
public static final int countryEstonia
```

countryFaroe_Denmark

```
public static final int countryFaroe_Denmark
```

countryFinland

```
public static final int countryFinland
```

countryFrance

```
public static final int countryFrance
```

countryGermany1

```
public static final int countryGermany1
```

countryGermany2

```
public static final int countryGermany2
```

countryGibraltar_UnitedKingdom

```
public static final int countryGibraltar_UnitedKingdom
```

countryGreece

```
public static final int countryGreece
```

countryHungary

```
public static final int countryHungary
```

countryIceland

```
public static final int countryIceland
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

countryIraq

```
public static final int countryIraq
```

countryIreland

```
public static final int countryIreland
```

countryIsrael

```
public static final int countryIsrael
```

countryItaly

```
public static final int countryItaly
```

countryJordan

```
public static final int countryJordan
```

countryLatvia

```
public static final int countryLatvia
```

countryLebanon

```
public static final int countryLebanon
```

countryLibya

```
public static final int countryLibya
```

countryLiechtenstein

```
public static final int countryLiechtenstein
```

countryLithuania

```
public static final int countryLithuania
```

countryLuxembourg

```
public static final int countryLuxembourg
```

countryMacedonia

```
public static final int countryMacedonia
```

countryMadeira_Portugal

```
public static final int countryMadeira_Portugal
```

countryMalta

```
public static final int countryMalta
```

countryMoldova

```
public static final int countryMoldova
```

countryMonaco

```
public static final int countryMonaco
```

countryMorocco

```
public static final int countryMorocco
```

countryNetherlands

```
public static final int countryNetherlands
```

countryNorways

```
public static final int countryNorways
```

countryPalestine

```
public static final int countryPalestine
```

countryPoland

```
public static final int countryPoland
```

countryPortugal

```
public static final int countryPortugal
```

countryRomania

```
public static final int countryRomania
```

countryRussianFederation

```
public static final int countryRussianFederation
```

countrySanMarino

```
public static final int countrySanMarino
```

countrySlovakia

```
public static final int countrySlovakia
```

countrySlovenia

```
public static final int countrySlovenia
```

countrySpain

```
public static final int countrySpain
```

countrySweden

```
public static final int countrySweden
```

countrySwitzerland

```
public static final int countrySwitzerland
```

countrySyrianArabRepublic

```
public static final int countrySyrianArabRepublic
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

countryTunisia

```
public static final int countryTunisia
```

countryTurkey

```
public static final int countryTurkey
```

countryUkraine

```
public static final int countryUkraine
```

countryUnitedKingdom

```
public static final int countryUnitedKingdom
```

countryVaticanCityState

```
public static final int countryVaticanCityState
```

countryYugoslavia

```
public static final int countryYugoslavia
```

acsNone

```
public static final int acsNone
```

acsNR_MSK

```
public static final int acsNR_MSK
```

acsEuroCryptEN50094

```
public static final int acsEuroCryptEN50094
```

acsReserverd1

```
public static final int acsReserverd1
```

acsReserverd2

```
public static final int acsReserverd2
```

acsReserverd3

```
public static final int acsReserverd3
```

acsReserverd4

```
public static final int acsReserverd4
```

acsReserverd5

```
public static final int acsReserverd5
```

componentTypeUnspecified

```
public static final int componentTypeUnspecified
```

componentTypeForegroundSound

```
public static final int componentTypeForegroundSound
```

componentTypeBackgroundSound

```
public static final int componentTypeBackgroundSound
```

componentTypeMultichannelAudio

```
public static final int componentTypeMultichannelAudio
```

componentTypeTrafficMessageChannel

```
public static final int componentTypeTrafficMessageChannel
```

componentTypeEmergencyWarningSystem

```
public static final int componentTypeEmergencyWarningSystem
```

componentTypeInteractiveTextTransmissionSystem

```
public static final int componentTypeInteractiveTextTransmissionSystem
```

componentTypePaging

```
public static final int componentTypePaging
```

componentTypeDynamicLabel

```
public static final int componentTypeDynamicLabel
```

componentTypeSlideshow

```
public static final int componentTypeSlideshow
```

componentTypeBroadcastWebSite

```
public static final int componentTypeBroadcastWebSite
```

componentTypeJava

```
public static final int componentTypeJava
```

componentTypeIPTunneling

```
public static final int componentTypeIPTunneling
```

languageUnkown

```
public static final int languageUnkown
```

languageAlbanian

```
public static final int languageAlbanian
```

languageBreton

```
public static final int languageBreton
```

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

languageCatalan

```
public static final int languageCatalan
```

languageCroatian

```
public static final int languageCroatian
```

languageWelsh

```
public static final int languageWelsh
```

languageCzech

```
public static final int languageCzech
```

languageDanish

```
public static final int languageDanish
```

languageGerman

```
public static final int languageGerman
```

languageEnglish

```
public static final int languageEnglish
```

languageSpanish

```
public static final int languageSpanish
```

languageEsperanto

```
public static final int languageEsperanto
```

languageEstonian

```
public static final int languageEstonian
```

languageBasque

```
public static final int languageBasque
```

languageFaroese

```
public static final int languageFaroese
```

languageFrench

```
public static final int languageFrench
```

languageFrisian

```
public static final int languageFrisian
```

languageIrish

```
public static final int languageIrish
```

languageGaelic

```
public static final int languageGaelic
```

languageGalician

```
public static final int languageGalician
```

languageIcelandic

```
public static final int languageIcelandic
```

languageItalian

```
public static final int languageItalian
```

languageLappish

```
public static final int languageLappish
```

languageLatin

```
public static final int languageLatin
```

languageLatvian

```
public static final int languageLatvian
```

languageLuxembourgian

```
public static final int languageLuxembourgian
```

languageLithuanian

```
public static final int languageLithuanian
```

languageHungarian

```
public static final int languageHungarian
```

languageMaltese

```
public static final int languageMaltese
```

languageDutch

```
public static final int languageDutch
```

languageNorwegian

```
public static final int languageNorwegian
```

languageOccitan

```
public static final int languageOccitan
```

languagePolish

```
public static final int languagePolish
```

languagePortuguese

```
public static final int languagePortuguese
```

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

languageRomanian

```
public static final int languageRomanian
```

languageRomansh

```
public static final int languageRomansh
```

languageSerbian

```
public static final int languageSerbian
```

languageSlovak

```
public static final int languageSlovak
```

languageSlovene

```
public static final int languageSlovene
```

languageFinnish

```
public static final int languageFinnish
```

languageSwedish

```
public static final int languageSwedish
```

languageTurkish

```
public static final int languageTurkish
```

languageFlemish

```
public static final int languageFlemish
```

languageWalloon

```
public static final int languageWalloon
```

language2C

```
public static final int language2C
```

language2D

```
public static final int language2D
```

language2E

```
public static final int language2E
```

language2F

```
public static final int language2F
```

languageNational30

```
public static final int languageNational30
```

languageNational31

```
public static final int languageNational31
```

languageNational32

```
public static final int languageNational32
```

languageNational33

```
public static final int languageNational33
```

languageNational34

```
public static final int languageNational34
```

languageNational35

```
public static final int languageNational35
```

languageNational36

```
public static final int languageNational36
```

languageNational37

```
public static final int languageNational37
```

languageNational38

```
public static final int languageNational38
```

languageNational39

```
public static final int languageNational39
```

languageNational3A

```
public static final int languageNational3A
```

languageNational3B

```
public static final int languageNational3B
```

languageNational3C

```
public static final int languageNational3C
```

languageNational3D

```
public static final int languageNational3D
```

languageNational3E

```
public static final int languageNational3E
```

languageNational3F

```
public static final int languageNational3F
```

languageAmharic

```
public static final int languageAmharic
```

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

languageArabic

```
public static final int languageArabic
```

languageArmenian

```
public static final int languageArmenian
```

languageAssamese

```
public static final int languageAssamese
```

languageAzerbaijani

```
public static final int languageAzerbaijani
```

languageBambora

```
public static final int languageBambora
```

languageBelorussian

```
public static final int languageBelorussian
```

languageBengali

```
public static final int languageBengali
```

languageBulgarian

```
public static final int languageBulgarian
```

languageBurmese

```
public static final int languageBurmese
```

languageChinese

```
public static final int languageChinese
```

languageChurash

```
public static final int languageChurash
```

languageDari

```
public static final int languageDari
```

languageFulani

```
public static final int languageFulani
```

languageGeorgian

```
public static final int languageGeorgian
```

languageGreek

```
public static final int languageGreek
```

languageGujurati

```
public static final int languageGujurati
```

languageGurani

```
public static final int languageGurani
```

languageHausa

```
public static final int languageHausa
```

languageHebrew

```
public static final int languageHebrew
```

languageHindi

```
public static final int languageHindi
```

languageIndonesian

```
public static final int languageIndonesian
```

languageJapanese

```
public static final int languageJapanese
```

languageKannada

```
public static final int languageKannada
```

languageKazakh

```
public static final int languageKazakh
```

languageKhmer

```
public static final int languageKhmer
```

languageKorean

```
public static final int languageKorean
```

languageLaotian

```
public static final int languageLaotian
```

languageMacedonian

```
public static final int languageMacedonian
```

languageMalagasay

```
public static final int languageMalagasay
```

languageMalaysian

```
public static final int languageMalaysian
```

languageMoldavian

```
public static final int languageMoldavian
```

languageMarathi

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

```
public static final int languageMarathi
```

languageNdebele

```
public static final int languageNdebele
```

languageNepali

```
public static final int languageNepali
```

languageOriya

```
public static final int languageOriya
```

languagePapamiento

```
public static final int languagePapamiento
```

languagePersian

```
public static final int languagePersian
```

languagePunjabi

```
public static final int languagePunjabi
```

languagePushtu

```
public static final int languagePushtu
```

languageQuechua

```
public static final int languageQuechua
```

languageRussian

```
public static final int languageRussian
```

languageRuthenian

```
public static final int languageRuthenian
```

languageSerbo_Croat

```
public static final int languageSerbo_Croat
```

languageShona

```
public static final int languageShona
```

languageSinhalese

```
public static final int languageSinhalese
```

languageSomali

```
public static final int languageSomali
```

languageSranan_Tongo

```
public static final int languageSranan_Tongo
```

languageSwahili

```
public static final int languageSwahili
```

languageTadzhik

```
public static final int languageTadzhik
```

languageTamil

```
public static final int languageTamil
```

languageTatar

```
public static final int languageTatar
```

languageTelugu

```
public static final int languageTelugu
```

languageThai

```
public static final int languageThai
```

languageUkrainian

```
public static final int languageUkrainian
```

languageUrdu

```
public static final int languageUrdu
```

languageUzbek

```
public static final int languageUzbek
```

languageVietnamese

```
public static final int languageVietnamese
```

languageZulu

```
public static final int languageZulu
```

language44

```
public static final int language44
```

language43

```
public static final int language43
```

language42

```
public static final int language42
```

language41

```
public static final int language41
```

languageBackgroundSoundCleanFeed

```
public static final int languageBackgroundSoundCleanFeed
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

charsetCompleteEBULatin

```
public static final int charsetCompleteEBULatin
```

charsetEBUCyrillicGreek

```
public static final int charsetEBUCyrillicGreek
```

charsetEBUArabic_HebrewETC

```
public static final int charsetEBUArabic_HebrewETC
```

charsetISOLatinAlphabetNo2

```
public static final int charsetISOLatinAlphabetNo2
```

serviceElementTypeUndefined

```
public static final int serviceElementTypeUndefined
```

serviceElementTypeEnsemble

```
public static final int serviceElementTypeEnsemble
```

serviceElementTypeService

```
public static final int serviceElementTypeService
```

serviceElementTypeComponent

```
public static final int serviceElementTypeComponent
```

locationInfoOnce

```
public static final int locationInfoOnce
```

locationInfoPeriodByTime

```
public static final int locationInfoPeriodByTime
```

locationInfoPeriodByDistance

```
public static final int locationInfoPeriodByDistance
```

locationInfoStop

```
public static final int locationInfoStop
```

locationInfoPosition

```
public static final int locationInfoPosition
```

locationInfoRegionId

```
public static final int locationInfoRegionId
```

operationControlSetVolume

```
public static final int operationControlSetVolume
```

operationControlGetVolume

```
public static final int operationControlGetVolume
```

operationControlSetServiceFollowing

```
public static final int operationControlSetServiceFollowing
```

operationControlGetServiceFollowing

```
public static final int operationControlGetServiceFollowing
```

operationControlGetServiceFollowingNotifications

```
public static final int operationControlGetServiceFollowingNotifications
```

operationControlSetDRCMode

```
public static final int operationControlSetDRCMode
```

operationControlGetDRCMode

```
public static final int operationControlGetDRCMode
```

operationControlGetDRCModeNotifications

```
public static final int operationControlGetDRCModeNotifications
```

serviceFollowingLeavingService

```
public static final int serviceFollowingLeavingService
```

serviceFollowingTryingAlternativeService

```
public static final int serviceFollowingTryingAlternativeService
```

serviceFollowingSelectingService

```
public static final int serviceFollowingSelectingService
```

streamTypeAudio

```
public static final int streamTypeAudio
```

streamTypePacket

```
public static final int streamTypePacket
```

streamTypeStream

```
public static final int streamTypeStream
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

streamTypeXPAD

```
public static final int streamTypeXPAD
```

streamTypeFIDC

```
public static final int streamTypeFIDC
```

subscriberInfoNoCA

```
public static final int subscriberInfoNoCA
```

subscriberInfoNoAlgorithm

```
public static final int subscriberInfoNoAlgorithm
```

subscriberInfoNoSubscription

```
public static final int subscriberInfoNoSubscription
```

subscriberInfoExpiredSubscription

```
public static final int subscriberInfoExpiredSubscription
```

conflictResolutionTurnProceed

```
public static final int conflictResolutionTurnProceed
```

conflictResolutionTurnProbe

```
public static final int conflictResolutionTurnProbe
```

conflictResolutionTurnStop

```
public static final int conflictResolutionTurnStop
```

conflictResolutionTurnPreempt

```
public static final int conflictResolutionTurnPreempt
```

conflictResolutionOperationNone

```
public static final int conflictResolutionOperationNone
```

conflictResolutionOperationTuneReq

```
public static final int conflictResolutionOperationTuneReq
```

conflictResolutionOperationSearchReq

```
public static final int conflictResolutionOperationSearchReq
```

conflictResolutionOperationScanReq

```
public static final int conflictResolutionOperationScanReq
```

conflictResolutionOperationSelectSIReq

```
public static final int conflictResolutionOperationSelectSIReq
```

conflictResolutionOperationGetEnsembleInfoReq

```
public static final int conflictResolutionOperationGetEnsembleInfoReq
```

conflictResolutionOperationGetServiceInfoReq

```
public static final int conflictResolutionOperationGetServiceInfoReq
```

conflictResolutionOperationGetComponentInfoReq

```
public static final int conflictResolutionOperationGetComponentInfoReq
```

conflictResolutionOperationSelectReceptionInfoReq

```
public static final int conflictResolutionOperationSelectReceptionInfoReq
```

conflictResolutionOperationSelectComponentReq

```
public static final int conflictResolutionOperationSelectComponentReq
```

conflictResolutionOperationSelectComponentStreamReq

```
public static final int conflictResolutionOperationSelectComponentStreamReq
```

conflictResolutionOperationSelectObjectReq

```
public static final int conflictResolutionOperationSelectObjectReq
```

conflictResolutionOperationGetLocationInfoReq

```
public static final int conflictResolutionOperationGetLocationInfoReq
```

conflictResolutionOperationOperationControlReq

```
public static final int conflictResolutionOperationOperationControlReq
```

conflictResolutionOperationSelectApplicationReq

```
public static final int conflictResolutionOperationSelectApplicationReq
```

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

conflictResolutionSuboperationNone

```
public static final int conflictResolutionSuboperationNone
```

conflictResolutionAnswerNo

```
public static final int conflictResolutionAnswerNo
```

conflictResolutionAnswerYes

```
public static final int conflictResolutionAnswerYes
```

Constructor Detail**DABConstants**

```
public DABConstants()
```

Method Detail**result2String**

```
public static java.lang.String result2String(int result)
```

The method returns a string for the given result code. It is an textual explanation of the result.

Class DABConnectionException[dab](#)

```
java.lang.Object
```

```
|
```

```
+- java.lang.Throwable
```

```
|
```

```
+- java.lang.Exception
```

```
|
```

```
+- dab.DABException
```

```
|
```

```
+- dab.DABConnectionException
```

All Implemented Interfaces:

```
java.io.Serializable
```

```
public Class DABConnectionException
```

```
extends DABException
```

The *DABConnectionException* is thrown when there are problems with the connection between the DAB client and the receiver.

Version:

1.0

Constructor Summary	Page
DABConnectionException()	334

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

DABConnectionException

```
public DABConnectionException()
```

Class DABClient

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

dab

```
java.lang.Object
|
+--dab.DABClient
```

All Implemented Interfaces:

[DABSource](#)

```
public Class DABClient
```

```
extends java.lang.Object
```

```
implements DABSource
```

The *DABClient* class is used to access a DAB resource. Usually the DAB resource might be a receiver that resides on the same host or is at least directly connected to it. But, it could also be a network device.

Note: the actual interface is defined in *DABSource*.

Version:

1.05

See Also:

[DABSource](#)

Constructor Summary		Page
DABClient ()	Create a DABClient object	336

Method Summary		Page
synchronized void	addDABListener (dab.DABListener listener) Register a DAB listener.	336
void	close () The connection to the current receiver is closed.	337
void	getComponentInfoReq (dab.si.ComponentId id)	338
void	getEnsembleInfoReq (dab.si.EnsembleId id)	338
void	getLocationInfoReq (int type, int mode, int desiredDelta, int desiredAccuracy)	340
void	getServiceInfoReq (dab.si.ServiceId id)	338
void	open () A connection to the default receiver is opened.	336
void	open (dab.DABReceiverAddress receiverAddress) A connection to the given receiver is opened.	337
void	operationControlReq (int attribute, java.lang.Object value)	340
synchronized void	removeDABListener (dab.DABListener listener) Removes the given listener from the list of DAB Listeners.	336
void	respondConflictResolutionReq (int transaction, int turn, int operation, int suboperation, int answer)	340
void	scanReq (int searchMode, int tables, int startFrequency, int stopFrequency, int transmissionModes, int notifications)	338
void	searchReq (int searchMode, int tables, int startFrequency, int stopFrequency, int transmissionModes, int notifications)	337
void	selectApplicationReq (dab.si.ComponentId componentId, dab.data.ObjectId objectId)	339
void	selectComponentReq (dab.si.ComponentId id, int selectionMode)	339
void	selectComponentStreamReq (dab.si.ComponentId componentId)	339
void	selectObjectReq (dab.si.ComponentId id, dab.data.ObjectId objectId, int requestMode, boolean replaceSelections, int deliveryMode, int cacheHint)	339
void	selectReceptionInfoReq (boolean synchronizationNotification, boolean bitErrorRateNotifications, boolean muteStateNotifications, boolean requestOnce)	339
void	selectSIReq (boolean ensembleInfo, boolean serviceInfo, boolean componentInfo, boolean autoDelivery)	338
void	tuneReq (int tuneFrequency, int transmissionMode)	337

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface `dab.DABSource`

[getComponentInfoReq](#), [getEnsembleInfoReq](#), [getLocationInfoReq](#), [getServiceInfoReq](#), [operationControlReq](#), [respondConflictResolutionReq](#), [scanReq](#), [searchReq](#), [selectApplicationReq](#), [selectComponentReq](#), [selectComponentStreamReq](#), [selectObjectReq](#), [selectReceptionInfoReq](#), [selectSIRReq](#), [tuneReq](#)

Constructor Detail

DABClient

```
public DABClient()
```

Create a `DABClient` object

Method Detail

addDABListener

```
public synchronized void addDABListener(dab.DABListener listener)
```

Register a DAB listener. DAB events, that relate to this client, are distributed to all registered listeners.

See Also:

[DABEvent](#), [DABListener](#), [removeDABListener](#)

removeDABListener

```
public synchronized void removeDABListener(dab.DABListener listener)
```

Removes the given listener from the list of DAB Listeners.

See Also:

[DABListener](#), [addDABListener](#)

open

```
public void open()  
    throws DABException,  
           java.lang.SecurityException
```

A connection to the default receiver is opened.

Throws:

[DABException](#) - when the client could not be registered

`SecurityException` - when the application controlling the `DABClient` does not have the permission to call the `open` method

open

```
public void open(dab.DABReceiverAddress receiverAddress)
    throws DABException,
           java.lang.SecurityException
```

A connection to the given receiver is opened. This method is only supported in configurations with multiple receivers.

Parameters:

`receiverAddress` - This parameter specifies the address of the receiver to be used

Throws:

[DABException](#) - when the client could not be registered

`SecurityException` - when the application controlling the *DABClient* does not have the permission to call the open method

close

```
public void close()
    throws DABException
```

The connection to the current receiver is closed. All ongoing transactions of the client are canceled.

Throws:

[DABException](#) - when no connection was opened

tuneReq

```
public void tuneReq(int tuneFrequency,
                   int transmissionMode)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

[tuneReq](#) in interface [DABSource](#)

searchReq

```
public void searchReq(int searchMode,
                     int tables,
                     int startFrequency,
                     int stopFrequency,
                     int transmissionModes,
                     int notifications)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[searchReq](#) in interface [DABSource](#)

scanReq

```
public void scanReq(int searchMode,
                   int tables,
                   int startFrequency,
                   int stopFrequency,
                   int transmissionModes,
                   int notifications)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[scanReq](#) in interface [DABSource](#)

selectSIReq

```
public void selectSIReq(boolean ensembleInfo,
                        boolean serviceInfo,
                        boolean componentInfo,
                        boolean autoDelivery)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[selectSIReq](#) in interface [DABSource](#)

getEnsembleInfoReq

```
public void getEnsembleInfoReq(dab.si.EnsembleId id)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[getEnsembleInfoReq](#) in interface [DABSource](#)

getServiceInfoReq

```
public void getServiceInfoReq(dab.si.ServiceId id)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[getServiceInfoReq](#) in interface [DABSource](#)

getComponentInfoReq

```
public void getComponentInfoReq(dab.si.ComponentId id)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[getComponentInfoReq](#) in interface [DABSource](#)

selectReceptionInfoReq

```
public void selectReceptionInfoReq(boolean synchronizationNotification,
                                     boolean bitErrorRateNotifications,
                                     boolean muteStateNotifications,
                                     boolean requestOnce)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[selectReceptionInfoReq](#) in interface [DABSource](#)

selectComponentReq

```
public void selectComponentReq(dab.si.ComponentId id,
                                 int selectionMode)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[selectComponentReq](#) in interface [DABSource](#)

selectComponentStreamReq

```
public void selectComponentStreamReq(dab.si.ComponentId componentId)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[selectComponentStreamReq](#) in interface [DABSource](#)

selectApplicationReq

```
public void selectApplicationReq(dab.si.ComponentId componentId,
                                   dab.data.ObjectId objectId)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

[selectApplicationReq](#) in interface [DABSource](#)

selectObjectReq

```
public void selectObjectReq(dab.si.ComponentId id,
                              dab.data.ObjectId objectId,
                              int requestMode,
                              boolean replaceSelections,
                              int deliveryMode,
                              int cacheHint)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[selectObjectReq](#) in interface [DABSource](#)

getLocationInfoReq

```
public void getLocationInfoReq(int type,
                               int mode,
                               int desiredDelta,
                               int desiredAccuracy)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[getLocationInfoReq](#) in interface [DABSource](#)

respondConflictResolutionReq

```
public void respondConflictResolutionReq(int transaction,
                                         int turn,
                                         int operation,
                                         int suboperation,
                                         int answer)
```

Specified by:

[respondConflictResolutionReq](#) in interface [DABSource](#)

operationControlReq

```
public void operationControlReq(int attribute,
                                java.lang.Object value)
    throws DABException,
           java.lang.SecurityException
```

Specified by:

[operationControlReq](#) in interface [DABSource](#)

Interface DABAppProxy

[dab](#)

All Superinterfaces:

[AppStateChangeEventSource](#)

public Interface **DABAppProxy**

extends [AppStateChangeEventSource](#)

This interface can be used to control applications that were launched using `selectApplicationReq` in `DABSource`.

Version:

0.2

See Also:

"Digital Video Broadcasting (DVB) Multimedia Home Platform (MHP), ETSI TS 101 812"

Field Summary		Page
int	<u>DESTROYED</u> the final state of the application - no further actions are possible	342
int	<u>INITED</u> the application is loaded and initialized, but there is no activity yet	342
int	<u>LOADED</u> the application is loaded	342
int	<u>NOT_LOADED</u> the application was selected, but is not yet loaded	342
int	<u>PAUSED</u> the application is paused, which means it is not active	342
int	<u>STARTED</u> the application is active	342

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
void	<u>addAppStateChangeListener</u> (dab.AppStateChangeListener listener) adds a listener for application state changes	343
java.lang.Object	<u>getAppProperty</u> (java.lang.String key) gets a property of the application.	345
int	<u>getState</u> () returns the current state of the application (cf. the defined constants)	343
void	<u>init</u> () initialises the application.	343
void	<u>load</u> () loads the classes of the application.	343
void	<u>pause</u> () pauses the application.	344
void	<u>removeAppStateChangeListener</u> (dab.AppStateChangeListener listener) removes a listener for application state changes	343
void	<u>resume</u> () resumes the application.	344

void	setAppProperty (java.lang.String key, java.lang.Object value)	345
	sets a property of the application.	
void	start ()	344
	starts the application.	
void	stop (boolean forced)	344
	requests to stop the application.	

Field Detail

DESTROYED

```
public static final int DESTROYED
```

the final state of the application - no further actions are possible

NOT_LOADED

```
public static final int NOT_LOADED
```

the application was selected, but is not yet loaded

LOADED

```
public static final int LOADED
```

the application is loaded

INITED

```
public static final int INITED
```

the application is loaded and initialized, but there is no activity yet

PAUSED

```
public static final int PAUSED
```

the application is paused, which means it is not active

STARTED

```
public static final int STARTED
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

the application is active

Method Detail**addAppStateChangeListener**

```
public void addAppStateChangeListener(dab.AppStateChangeListener listener)
```

adds a listener for application state changes

removeAppStateChangeListener

```
public void removeAppStateChangeListener(dab.AppStateChangeListener listener)
```

removes a listener for application state changes

getState

```
public int getState()  
    throws java.lang.SecurityException
```

returns the current state of the application (cf. the defined constants)

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to retrieve the application state

load

```
public void load()  
    throws java.lang.SecurityException
```

loads the classes of the application. The state of the application changes to LOADED.

This action is only successful, if the application was not loaded before.

A state change event is signalled in any case.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to load the application

init

```
public void init()  
    throws java.lang.SecurityException
```

initialises the application. The routine `initXlet` in the related application will be called. The application is afterwards in the INITED state.

This action is only successful, if the application was not initialised before. If the application was not loaded, the application will first be loaded and then initialised.

A state change event is signalled in any case. An additional state change is signalled if the application also has to be loaded. In this case first the state change to LOADED is signalled and afterwards that one to INITED.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to initialise the application

See Also:

[Xlet](#)

pause

```
public void pause()  
    throws java.lang.SecurityException
```

pauses the application. The routine `pauseXlet` in the related application will be called. The application is afterwards in the PAUSED state.

This action is only successful, if the application is either in the INITED state or in the STARTED state.

A state change event is signalled in any case.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to pause the application

See Also:

[Xlet](#)

resume

```
public void resume()  
    throws java.lang.SecurityException
```

resumes the application. The routine `startXlet` in the related application will be called. The application is afterwards in the STARTED state.

This action is only successful, if the application was in the PAUSED state.

A state change event is signalled in any case.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to resume the application

See Also:

[Xlet](#)

start

```
public void start()  
    throws java.lang.SecurityException
```

starts the application. The routine `startXlet` in the related application will be called. The application is afterwards in the STARTED state.

This action is only successful, if the application was not paused or destroyed. If the application was not loaded, the application will first be loaded, then initialised and finally be started. If the application was not initialised, the application will be initialised and then started.

A state change event is signalled in any case. Additional state changes will also be signalled (e.g. NOT_LOADED - LOADED - INITED - STARTED).

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to start the application

See Also:

[Xlet](#)

stop

```
public void stop(boolean forced)  
    throws java.lang.SecurityException
```

requests to stop the application. The routine `destroyXlet` in the related application will be called. The application is afterwards in the DESTROYED state.

This action is only successful, if the application was not destroyed.

A state change event is signalled in any case.

Parameters:

forced - if set to true the application is asked to stop and may refuse. if set to false, the application is stopped in any case.

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to stop the application

See Also:

[Xlet](#)

setAppProperty

```
public void setAppProperty(java.lang.String key,  
                           java.lang.Object value)  
    throws java.lang.SecurityException
```

sets a property of the application.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Parameters:

key - the name of the property
value - the new value of the property

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to set an property of the application

getAppProperty

```
public java.lang.Object getAppProperty(java.lang.String key)  
    throws java.lang.SecurityException
```

gets a property of the application. It returns the value of the property or NULL if the property is not defined.

Parameters:

key - the name of the property

Throws:

`java.lang.SecurityException` - the exception is thrown if the caller is not permitted to retrieve an property of the application

Class DABAdapter

[dab](#)

```
java.lang.Object
```

```
|
```

```
+-dab.DABAdapter
```

All Implemented Interfaces:

[DABListener](#)

```
public Class DABAdapter
```

```
extends java.lang.Object
```

```
implements DABListener
```

The *DABAdapter* class provides default methods for the implementation of a *DABListener*.
The default behavior is that incoming events are ignored.

Version:

1.01

See Also:[DABListener](#)

Constructor Summary	Page
DABAdapter ()	347

Method Summary	Page
void componentNtf (dab.events.ComponentNtfEvent e)	349
void conflictResolutionNtf (dab.events.ConflictResolutionNtfEvent e)	350
void drcModeNtf (dab.events.DRCModeNtfEvent e)	350
void getComponentInfoCnf (dab.events.GetComponentInfoCnfEvent e)	348
void getEnsembleInfoCnf (dab.events.GetEnsembleInfoCnfEvent e)	348
void getLocationInfoCnf (dab.events.GetLocationInfoCnfEvent e)	350
void getServiceInfoCnf (dab.events.GetServiceInfoCnfEvent e)	348
void locationInfoNtf (dab.events.LocationInfoNtfEvent e)	350
void objectNtf (dab.events.ObjectNtfEvent e)	349
void operationControlCnf (dab.events.OperationControlCnfEvent e)	350
void receptionInfoNtf (dab.events.ReceptionInfoNtfEvent e)	349
void respondConflictResolutionCnf (dab.events.RespondConflictResolutionCnfEvent e)	350
void scanCnf (dab.events.ScanCnfEvent e)	347
void scanNtf (dab.events.ScanNtfEvent e)	347
void searchCnf (dab.events.SearchCnfEvent e)	347
void searchNtf (dab.events.SearchNtfEvent e)	347
void selectApplicationCnf (dab.events.SelectApplicationCnfEvent e)	349
void selectComponentCnf (dab.events.SelectComponentCnfEvent e)	349
void selectComponentStreamCnf (dab.events.SelectComponentStreamCnfEvent e)	349
void selectObjectCnf (dab.events.SelectObjectCnfEvent e)	349
void selectReceptionInfoCnf (dab.events.SelectReceptionInfoCnfEvent e)	348
void selectSICnf (dab.events.SelectSICnfEvent e)	348
void serviceFollowingNtf (dab.events.ServiceFollowingNtfEvent e)	350
void siNtf (dab.events.SINtfEvent e)	348
void systemFailureNtf (dab.events.SystemFailureNtfEvent e)	351
void tuneCnf (dab.events.TuneCnfEvent e)	347

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface dab.[DABListener](#)

[componentNtf](#), [conflictResolutionNtf](#), [drcModeNtf](#), [getComponentInfoCnf](#), [getEnsembleInfoCnf](#), [getLocationInfoCnf](#), [getServiceInfoCnf](#), [locationInfoNtf](#), [objectNtf](#), [operationControlCnf](#), [receptionInfoNtf](#), [respondConflictResolutionCnf](#), [scanCnf](#), [scanNtf](#), [searchCnf](#), [searchNtf](#), [selectApplicationCnf](#), [selectComponentCnf](#), [selectComponentStreamCnf](#), [selectObjectCnf](#), [selectReceptionInfoCnf](#), [selectSICnf](#), [serviceFollowingNtf](#), [siNtf](#), [systemFailureNtf](#), [tuneCnf](#)

Constructor Detail**DABAdapter**

```
public DABAdapter()
```

Method Detail**tuneCnf**

```
public void tuneCnf(dab.events.TuneCnfEvent e)
```

Specified by:

[tuneCnf](#) in interface [DABListener](#)

searchCnf

```
public void searchCnf(dab.events.SearchCnfEvent e)
```

Specified by:

[searchCnf](#) in interface [DABListener](#)

searchNtf

```
public void searchNtf(dab.events.SearchNtfEvent e)
```

Specified by:

[searchNtf](#) in interface [DABListener](#)

scanCnf

```
public void scanCnf(dab.events.ScanCnfEvent e)
```

Specified by:

[scanCnf](#) in interface [DABListener](#)

scanNtf

```
public void scanNtf(dab.events.ScanNtfEvent e)
```

Specified by:

[scanNtf](#) in interface [DABListener](#)

selectSICnf

```
public void selectSICnf(dab.events.SelectSICnfEvent e)
```

Specified by:

[selectSICnf](#) in interface [DABListener](#)

siNtf

```
public void siNtf(dab.events.SINtfEvent e)
```

Specified by:

[siNtf](#) in interface [DABListener](#)

getEnsembleInfoCnf

```
public void getEnsembleInfoCnf(dab.events.GetEnsembleInfoCnfEvent e)
```

Specified by:

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

[getEnsembleInfoCnf](#) in interface [DABListener](#)

getServiceInfoCnf

```
public void getServiceInfoCnf(dab.events.GetServiceInfoCnfEvent e)
```

Specified by:

[getServiceInfoCnf](#) in interface [DABListener](#)

getComponentInfoCnf

```
public void getComponentInfoCnf(dab.events.GetComponentInfoCnfEvent e)
```

Specified by:

[getComponentInfoCnf](#) in interface [DABListener](#)

selectReceptionInfoCnf

```
public void selectReceptionInfoCnf(dab.events.SelectReceptionInfoCnfEvent e)
```

Specified by:

[selectReceptionInfoCnf](#) in interface [DABListener](#)

receptionInfoNtf

```
public void receptionInfoNtf(dab.events.ReceptionInfoNtfEvent e)
```

Specified by:

[receptionInfoNtf](#) in interface [DABListener](#)

selectComponentCnf

```
public void selectComponentCnf(dab.events.SelectComponentCnfEvent e)
```

Specified by:

[selectComponentCnf](#) in interface [DABListener](#)

selectComponentStreamCnf

```
public void selectComponentStreamCnf(dab.events.SelectComponentStreamCnfEvent e)
```

Specified by:

[selectComponentStreamCnf](#) in interface [DABListener](#)

componentNtf

```
public void componentNtf(dab.events.ComponentNtfEvent e)
```

Specified by:

[componentNtf](#) in interface [DABListener](#)

selectObjectCnf

```
public void selectObjectCnf(dab.events.SelectObjectCnfEvent e)
```

Specified by:

[selectObjectCnf](#) in interface [DABListener](#)

selectApplicationCnf

```
public void selectApplicationCnf(dab.events.SelectApplicationCnfEvent e)
```

Specified by:

[selectApplicationCnf](#) in interface [DABListener](#)

objectNtf

```
public void objectNtf(dab.events.ObjectNtfEvent e)
```

Specified by:

[objectNtf](#) in interface [DABListener](#)

getLocationInfoCnf

```
public void getLocationInfoCnf(dab.events.GetLocationInfoCnfEvent e)
```

Specified by:

[getLocationInfoCnf](#) in interface [DABListener](#)

locationInfoNtf

```
public void locationInfoNtf(dab.events.LocationInfoNtfEvent e)
```

Specified by:

[locationInfoNtf](#) in interface [DABListener](#)

conflictResolutionNtf

```
public void conflictResolutionNtf(dab.events.ConflictResolutionNtfEvent e)
```

Specified by:

[conflictResolutionNtf](#) in interface [DABListener](#)

respondConflictResolutionCnf

```
public void respondConflictResolutionCnf(dab.events.RespondConflictResolutionCnfEvent e)
```

Specified by:

[respondConflictResolutionCnf](#) in interface [DABListener](#)

operationControlCnf

```
public void operationControlCnf(dab.events.OperationControlCnfEvent e)
```

Specified by:

[operationControlCnf](#) in interface [DABListener](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

serviceFollowingNtf

```
public void serviceFollowingNtf(dab.events.ServiceFollowingNtfEvent e)
```

Specified by:

[serviceFollowingNtf](#) in interface [DABListener](#)

drcModeNtf

```
public void drcModeNtf(dab.events.DRCModeNtfEvent e)
```

Specified by:

[drcModeNtf](#) in interface [DABListener](#)

systemFailureNtf

```
public void systemFailureNtf(dab.events.SystemFailureNtfEvent e)
```

Specified by:

[systemFailureNtf](#) in interface [DABListener](#)

Interface AppStateChangeEventSource

[dab](#)

All Known Subinterfaces:

[DABAppProxy](#)

```
public Interface AppStateChangeEventSource
```

An interface that can be used to identify an application by its related controller.

Version:

0.2

Interface AppStateChangeListener

[dab](#)

```
public Interface AppStateChangeListener
```

AppStateChangeListener defines the interface for events originating from the DABAppProxy. Note: to support J2ME MIDP Profile AppStateChangeListener does not implement java.util.EventListener any more

Version:

0.3

See Also:

"Digital Video Broadcasting (DVB) Multimedia Home Platform (MHP), ETSI TS 101 812", [DABAppProxy](#)

Method Summary		Page
void	stateChange (dab.events.AppStateChangeEvent event)	351
	This method is used to signal a state change for the related application.	

Method Detail

stateChange

```
public void stateChange(dab.events.AppStateChangeEvent event)
```

This method is used to signal a state change for the related application.

Class **SubscriberInfo**

[dab.data](#)

java.lang.Object

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

|

+--**dab.data.SubscriberInfo**

public Class **SubscriberInfo**

extends java.lang.Object

SubscriberInfo contains information how to subscribe to a service.

Version:

1.00

Constructor Summary		Page
SubscriberInfo ()		352

Method Summary		Page
java.lang.String	getContentName () returns the content name of the alternative object	367
int	getEncryptionSpecificFlags () returns a flag field that can be used for redirection purposes	353
int	getReason () returns a flag field (cf.	452

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SubscriberInfo

public **SubscriberInfo**()

Method Detail**getReason**

```
public int getReason()
```

returns a flag field (cf. DABConstants.subscriberInfo*) that explains why the related BWS object could not be descrambled

getEncryptionSpecificFlags

```
public int getEncryptionSpecificFlags()
```

returns a flag field that can be used for redirection purposes

getContentName

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

returns the content name of the alternative object

Class ProgrammeType

[dab.data](#)

```
java.lang.Object
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

```
|
+--dab.data.ProgrammeType
```

```
public Class ProgrammeType
```

```
extends java.lang.Object
```

ProgrammeType represents provided programme types of a certain service. It consists of an international code, an optional coarse code and two optional fine codes.

Version:

1.01

Constructor Summary

Constructor Summary		<i>Page</i>
ProgrammeType (int internationalCode, byte[] fineCode, boolean hasCoarseCode, int coarseCode)		354

Method Summary

Method Summary		<i>Page</i>
int getCoarseCode ()	Returns the coarse code	354

byte[]	getFineCode() Returns a reference to the fine codes	354
int	getInternationalCode() Returns the international code	354

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail**ProgrammeType**

```
public ProgrammeType(int internationalCode,  
                    byte[] fineCode,  
                    boolean hasCoarseCode,  
                    int coarseCode)
```

Method Detail**getInternationalCode**

```
public int getInternationalCode()
```

Returns the international code

getCoarseCode

```
public int getCoarseCode()  
    throws DABNotAvailableException
```

Returns the coarse code

Throws:

[DABNotAvailableException](#) - when the code is not available

getFineCode

```
public byte[] getFineCode()
```

Returns a reference to the fine codes

Class ProgrammeNumber

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.ProgrammeNumber
```

```
public Class ProgrammeNumber
```

```
extends java.lang.Object
```

ProgrammeNumber represents a programme number that can be used for "programming" a service.

Version:

```
1.00
```

Constructor Summary		Page
ProgrammeNumber ()		355

Method Summary		Page
dab.si.ServiceId	getNewService () returns the ServiceId of the new service when the programme is redirected	356
java.util.Date	getTransmissionTime () returns the transmission time	355
boolean	isInterrupted () signals, whether the programme is interrupted by later continued	356
boolean	isRedirected () signals, whether the programme is redirected to a different service and time	356

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail**ProgrammeNumber**

```
public ProgrammeNumber( )
```

Method Detail**getTransmissionTime**

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

```
returns the transmission time
```

isInterrupted

```
public boolean isInterrupted()
```

signals, whether the programme is interrupted by later continued

isRedirected

```
public boolean isRedirected()
```

signals, whether the programme is redirected to a different service and time

See Also:

[getNewService](#)

getNewService

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

```
public dab.si.ServiceId getNewService()
```

returns the ServiceId of the new service when the programme is redirected

See Also:

[isRedirected](#)

Class ObjectId

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.ObjectId
```

```
public Class ObjectId
```

```
extends java.lang.Object
```

The **ObjectId** is an identifier for objects carried in a data service channel. It is used to request objects and for identification of delivered objects to the application.

Version:

1.04

Constructor Summary		Page
ObjectId ()	Constructs an ObjectId object	357
ObjectId (dab.data.ObjectId objectId)	Constructs a copy of the given ObjectId.	357
ObjectId (java.lang.String stringId)	Constructs an ObjectId object from the given string.	357

Method Summary		Page
int	compareTo (dab.data.ObjectId objectId) This method compares the object with the given object.	358
int	compareTo (java.lang.Object objectId) This method compares the object with the given object.	391
java.lang.String	getId () Returns an external representation of the identifier in a textual format.	403

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

ObjectId

```
public ObjectId()
```

Constructs an ObjectId object

ObjectId

```
public ObjectId(dab.data.ObjectId objectId)
```

Constructs a copy of the given ObjectId.

See Also:

[getId](#)

ObjectId

```
public ObjectId(java.lang.String stringId)
```

Constructs an ObjectId object from the given string.

See Also:

[getId](#)

Method Detail**compareTo**

```
public int compareTo(java.lang.Object objectId)
```

This method compares the object with the given object. The behavior is the same as it is specified in the compareTo method of the java.lang.Comparable interface.

compareTo

```
public int compareTo(dab.data.ObjectId objectId)
```

This method compares the object with the given object. The behavior is the same as it is specified in the compareTo method of the java.lang.Comparable interface.

getId

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

Returns an external representation of the identifier in a textual format. The returned value can be used to construct an object id.

Interface MOTObjectHeader

[dab.data](#)

All Known Implementing Classes:

[BWSDirectoryObject](#), [BWSObject](#), [MOTDirectoryObject](#), [MOTObject](#)

```
public Interface MOTObjectHeader
```

The MOTObjectHeader represents the header information of an MOT object

Note: parameter values are defined in EN 301 234 "Digital Audio Broadcasting (DAB): Multimedia Object Transfer protocol"

Version:

1.03

Method Summary		Page
int	getCompressionType () get the compression type of the object	370
java.lang.String	getContentDescription () Deprecated. <i>obsolete parameter as defined in EN 301 234</i>	366
int	getContentDescriptionCharset () Deprecated. <i>obsolete parameter as defined in EN 301 234</i>	367
java.lang.String	getContentName () get the content name	367

int	getContentNameCharset () get the charset of the content name (see DABConstants.charset*)	367
int	getContentSubtype () get the content subtype (the exact type) as defined in EN 301 234	366
int	getContentType () get the content type (the main category) as defined in EN 301 234	366
java.util.Date	getCreationTime () Deprecated. <i>obsolete parameter as defined in EN 301 234</i>	368
int	getExpiration () get the relative expiration time in minutes - if an absolute expire time is signalled expiration has to be transformed to a value in minutes - if an relative expire time is signalled the expiration is always be transformed in a value in minutes	363
java.util.Date	getExpireTime () Deprecated. <i>obsolete parameter as defined in EN 301 234, use getExpiration() instead</i>	369
dab.data.Label	getLabel () get the label	404
java.lang.String	getMimeType () Returns the MIME type of the object	370
int	getPriority () get the priority (0=lowest priority; 255=highest priority) NOTE: for MOT directory mode only	368
int	getRepetitionDistance () get the repetition distance (in 1/10s)	368
java.util.Date	getStartValidity () Deprecated. <i>obsolete parameter as defined in EN 301 234</i>	369
java.util.Date	getTriggerTime () the date for presenting the object	369
int	getUniqueBodyVersion () get the unique body version of the object	362
boolean	getValidity () returns false, if validity is now; otherwise true.	368
int	getVersionNumber () Deprecated. <i>obsolete parameter as defined in EN 301 234, use getUniqueBodyVersion() instead</i>	368

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

Method Detail**getContentType**

```
public int getContentType()
```

get the content type (the main category) as defined in EN 301 234

Returns:

MOT ContentType

getContentSubtype

```
public int getContentSubtype()
```

get the content subtype (the exact type) as defined in EN 301 234

Returns:

MOT ContentSubType

getContentDescription

```
public java.lang.String getContentDescription()  
throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234*

get the content description

Returns:

MOT ContentDescription

Throws:

[DABNotAvailableException](#) - when the content description is not available

getContentDescriptionCharset

```
public int getContentDescriptionCharset()  
throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234*

get the charset of the content description (see DABConstants.charset*)

Returns:

MOT charsetset of ContentDescription

Throws:

[DABNotAvailableException](#) - when the charset is not available

See Also:

DABConstants

getContentName

```
public java.lang.String getContentName()  
throws DABNotAvailableException
```

get the content name

Returns:

MOT ContentName

Throws:

[DABNotAvailableException](#) - when the content name is not available

getContentNameCharset

```
public int getContentNameCharset()  
        throws DABNotAvailableException
```

get the charset of the content name (see DABConstants.charset*)

Returns:

MOT ContentName Characterset

Throws:

[DABNotAvailableException](#) - when the charset is not available

See Also:

DABConstants

getLabel

```
public dab.data.Label getLabel()  
        throws DABNotAvailableException
```

get the label

Returns:

MOT Label

Throws:

[DABNotAvailableException](#) - when the label is not available

getPriority

```
public int getPriority()  
        throws DABNotAvailableException
```

get the priority (0=lowest priority; 255=highest priority)

NOTE: for MOT directory mode only

Returns:

MOT Priority as defined in as defined in EN 301 234

Throws:

[DABNotAvailableException](#) - when the priority is not available

getRepetitionDistance

```
public int getRepetitionDistance()  
        throws DABNotAvailableException
```

get the repetition distance (in 1/10s)

Returns:

MOT repetition distance in 1/10s

Throws:

[DABNotAvailableException](#) - when the repetition distance is not available

getVersionNumber

```
public int getVersionNumber()  
        throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234, use `getUniqueBodyVersion()` instead*

get the version number of the object

Returns:

MOT version number of object

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

Throws:

[DABNotAvailableException](#) - when the version number is not available

See Also:

[getUniqueBodyVersion](#)

getUniqueBodyVersion

```
public int getUniqueBodyVersion()  
        throws DABNotAvailableException
```

get the unique body version of the object

Returns:

MOT unique body version as defined in EN 301 234

Throws:

[DABNotAvailableException](#) - when the unique body version is not available

Since:

version 1.03

getValidity

```
public boolean getValidity()
```

returns false, if validity is now; otherwise true. Note, if the validity is set to false the referred time routines have to be ignored.

See Also:

[getCreationTime](#), [getStartValidity](#), [getExpireTime](#), [getTriggerTime](#)

getCreationTime

```
public java.util.Date getCreationTime()  
        throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234*

get the authoring date of the object

Returns:

the authoring date of the object

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getStartValidity

```
public java.util.Date getStartValidity()  
        throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234*

get the date after which the object is valid

Returns:

date after which the object is valid

Throws:

[DABNotAvailableException](#) - when not available

See Also:[getValidity](#)

getExpireTime

```
public java.util.Date getExpireTime()
    throws DABNotAvailableException
```

Deprecated. *obsolete parameter as defined in EN 301 234, use getExpiration() instead*

get the date after which the object is not valid anymore

Returns:

the date after which the object is not valid anymore

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#), [getExpiration](#)

getExpiration

```
public int getExpiration()
    throws DABNotAvailableException
```

get the relative expiration time in minutes

- if an absolute expire time is signalled expiration has to be transformed to a value in minutes - if an relative expire time is signalled the expiration is always be transformed in a value in minutes

Returns:

the relative expiration time in minutes

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getTriggerTime

```
public java.util.Date getTriggerTime()
    throws DABNotAvailableException
```

the date for presenting the object

Returns:

the date for presenting the object

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getMimeType

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

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

Returns the MIME type of the object

getCompressionType

```
public int getCompressionType()
    throws DABNotAvailableException
```

get the compression type of the object

Returns:

the compression type of the object as defined in EN 301 234

Throws:

[DABNotAvailableException](#) - when the content description is not available

Class MOTObject

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.DABObject
```

```
|
```

```
+--dab.data.MOTObject
```

All Implemented Interfaces:

[MOTObjectHeader](#)

Direct Known Subclasses:

[BWSObject](#), [MOTDirectoryObject](#)

```
public Class MOTObject
```

```
extends DABObject
```

```
implements MOTObjectHeader
```

The MOTObject represents data that is transported via the MOT protocol.

Version:

1.07

Constructor Summary	<i>Page</i>
MOTObject ()	366

Method Summary	<i>Page</i>
byte[] getBody ()	366
int getCompressionType ()	370
java.lang.String getContentDescription ()	366

int	<u>getContentDescriptionCharset</u> () Returns the charset of the content description (cf.	367
java.lang.String	<u>getContentName</u> () Returns the content name	367
int	<u>getContentNameCharset</u> () Returns the charset of the content name (cf.	367
int	<u>getContentSubtype</u> () Returns the content subtype (the exact type)	366
int	<u>getContentType</u> () Returns the content type (the main category)	366
java.util.Date	<u>getCreationTime</u> () Returns the authoring date of the object	368
java.util.Date	<u>getExpireTime</u> () Returns the date after which the object is not valid anymore	369
dab.data.Label	<u>getLabel</u> () Returns the label	404
java.lang.String	<u>getMimeType</u> () Returns the MIME type of the object	370
int	<u>getPriority</u> () Returns the priority (0=lowest priority; 255=highest priority)	368
int	<u>getRepetitionDistance</u> () Returns the repetition distance (in ms)	368
java.util.Date	<u>getStartValidity</u> () Returns the date after which the object is valid	369
java.util.Date	<u>getTriggerTime</u> () Returns the date for presenting the object	369
boolean	<u>getValidity</u> () Returns false, if validity is now; otherwise true.	368
int	<u>getVersionNumber</u> () Returns the version number of the object	368
java.lang.String	<u>toString</u> () Returns a textual representation of the object	389

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface dab.data.[MOTObjectHeader](#)

[getCompressionType](#), [getContentDescription](#), [getContentDescriptionCharset](#), [getContentName](#), [getContentNameCharset](#), [getContentSubtype](#), [getContentType](#), [getCreationTime](#), [getExpiration](#), [getExpireTime](#), [getLabel](#), [getMimeType](#), [getPriority](#), [getRepetitionDistance](#), [getStartValidity](#), [getTriggerTime](#), [getUniqueBodyVersion](#), [getValidity](#), [getVersionNumber](#)

Constructor Detail**MOTObject**

```
public MOTObject()
```

Method Detail**getContentType**

```
public int getContentType()
```

Returns the content type (the main category)

Specified by:

[getContentType](#) in interface [MOTObjectHeader](#)

getContentSubtype

```
public int getContentSubtype()
```

Returns the content subtype (the exact type)

Specified by:

[getContentSubtype](#) in interface [MOTObjectHeader](#)

getBody

```
public byte[] getBody()
```

Returns the body of the object (the actual content)

getContentDescription

```
public java.lang.String getContentDescription()  
throws DABNotAvailableException
```

Returns the content description

Specified by:

[getContentDescription](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the content description is not available

getContentDescriptionCharset

```
public int getContentDescriptionCharset()  
        throws DABNotAvailableException
```

Returns the charset of the content description (cf. DABConstants.charset*)

Specified by:

[getContentDescriptionCharset](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the charset is not available

getContentName

```
public java.lang.String getContentName()  
        throws DABNotAvailableException
```

Returns the content name

Specified by:

[getContentName](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the content name is not available

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

getContentNameCharset

```
public int getContentNameCharset()  
        throws DABNotAvailableException
```

Returns the charset of the content name (cf. DABConstants.charset*)

Specified by:

[getContentNameCharset](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the charset is not available

getLabel

```
public dab.data.Label getLabel()  
        throws DABNotAvailableException
```

Returns the label

Specified by:

[getLabel](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the priority is not available

getPriority

```
public int getPriority()  
        throws DABNotAvailableException
```

Returns the priority (0=lowest priority; 255=highest priority)

Specified by:

[getPriority](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the priority is not available

getRepetitionDistance

```
public int getRepetitionDistance()  
        throws DABNotAvailableException
```

Returns the repetition distance (in ms)

Specified by:

[getRepetitionDistance](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the repetition distance is not available

getVersionNumber

```
public int getVersionNumber()  
        throws DABNotAvailableException
```

Returns the version number of the object

Specified by:

[getVersionNumber](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the version is not available

getValidity

```
public boolean getValidity()
```

Returns false, if validity is now; otherwise true. Note, if the validity is set to false the referred time routines have to be ignored.

Specified by:

[getValidity](#) in interface [MOTObjectHeader](#)

See Also:

[getCreationTime](#), [getStartValidity](#), [getExpireTime](#), [getTriggerTime](#)

getCreationTime

```
public java.util.Date getCreationTime()  
        throws DABNotAvailableException
```

Returns the authoring date of the object

Specified by:

[getCreationTime](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getStartValidity

```
public java.util.Date getStartValidity()
                        throws DABNotAvailableException
```

Returns the date after which the object is valid

Specified by:

[getStartValidity](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

getExpireTime

```
public java.util.Date getExpireTime()
                        throws DABNotAvailableException
```

Returns the date after which the object is not valid anymore

Specified by:

[getExpireTime](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

getTriggerTime

```
public java.util.Date getTriggerTime()
                        throws DABNotAvailableException
```

Returns the date for presenting the object

Specified by:

[getTriggerTime](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when not available

See Also:

[getValidity](#)

toString

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

Returns a textual representation of the object

Overrides:

toString in class java.lang.Object

getMimeType

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

Returns the MIME type of the object

Specified by:

[getMimeType](#) in interface [MOTObjectHeader](#)

getCompressionType

```
public int getCompressionType()  
        throws DABNotAvailableException
```

Returns the compression type of the object

Specified by:

[getCompressionType](#) in interface [MOTObjectHeader](#)

Throws:

[DABNotAvailableException](#) - when the content description is not available

Class MOTDirectoryObject

[dab.data](#)

```
java.lang.Object
```

```
|  
+--dab.data.DABObject  
    |  
    +--dab.data.MOTObject  
        |  
        +--dab.data.MOTDirectoryObject
```

All Implemented Interfaces:

[MOTObjectHeader](#)

Direct Known Subclasses:

[BWSDirectoryObject](#)

```
public Class MOTDirectoryObject
```

```
extends MOTObject
```

The MOTDirectoryObject class represents a MOT carousel directory of a component

Version:

1.02

Constructor Summary		Page
MOTDirectoryObject ()		371

Method Summary		Page
int	getCarouselPeriod () get maximum time (in tenths of second) for the carousel to cycle	372
dab.data.MOTObjectHeader[]	getContents () Returns MOT Headers for objects described by the directory	372
int	getNumberOfObjects () get number of objects described by the directory	371

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class dab.data. MOTObject
getBody , getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContentType , getCreationTime , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getValidity , getVersionNumber , toString

Methods inherited from class java.lang.Object
equals , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Methods inherited from interface dab.data. MOTObjectHeader
getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContentType , getCreationTime , getExpiration , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getUniqueBodyVersion , getValidity , getVersionNumber

Constructor Detail

MOTDirectoryObject

```
public MOTDirectoryObject()
```

Method Detail

getNumberOfObjects

```
public int getNumberOfObjects()
```

get number of objects described by the directory

Returns:

number of objects in the directory

getCarouselPeriod

```
public int getCarouselPeriod()
```

get maximum time (in tenths of second) for the carousel to cycle

Returns:

carousel period

getContents

```
public dab.data.MOTObjectHeader[] getContents()
```

Returns MOT Headers for objects described by the directory

Returns:

array of mot objects in the directory

Class LocationInfo

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

dab.data

```
java.lang.Object
```

```
|
```

```
+--dab.data.LocationInfo
```

```
public Class LocationInfo
```

```
extends java.lang.Object
```

LocationInfo represents location data this is returned by the GetLocationInfo command.

Note, if the quality is below zero, than all other attributes are invalid.

The used coordinates have the same reference system as GPS.

Version:

1.01

Constructor Summary		Page
LocationInfo ()		373

Method Summary		Page
int getAltitude ()	Returns the altitude in meters above ground.	373

int	getDirection () Returns the direction in 100.000ths of a degree (range: [0,360[in degrees; 0 degrees points to north).	374
int	getLatitude () Returns the latitude in 100.000ths of a degree (from +90 degrees for northerly latitudes to -90 degrees for southerly latitudes).	373
int	getLongitude () Returns the longitude in 100.000ths of a degree (from +180 degrees for easterly longitudes to -180 degrees for westerly longitudes).	373
int	getQuality () Returns the overall quality of the data.	374
int	getVelocity () Returns the velocity in 100.000ths of a meter per second.	374

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

LocationInfo

```
public LocationInfo()
```

Method Detail

getLongitude

```
public int getLongitude()
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Returns the longitude in 100.000ths of a degree (from +180 degrees for easterly longitudes to -180 degrees for westerly longitudes).

getLatitude

```
public int getLatitude()
```

Returns the latitude in 100.000ths of a degree (from +90 degrees for northerly latitudes to -90 degrees for southerly latitudes).

getAltitude

```
public int getAltitude()
```

Returns the altitude in meters above ground.

getVelocity

```
public int getVelocity()
```

Returns the velocity in 100.000ths of a meter per second.

getDirection

```
public int getDirection()
```

Returns the direction in 100.000ths of a degree (range: [0,360[in degrees; 0 degrees points to north).

getQuality

```
public int getQuality()
```

Returns the overall quality of the data. The range is from +100 (best) to -100 (worst). Negative values indicates invalid data.

Class Label

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.Label
```

```
public Class Label
```

```
extends java.lang.Object
```

Label models a textual string which is used in the DAB System for service labels, object labels and so on. It contains a text with max. 16 characters. Additionally the character set is indicated and it is specified how the label is to be displayed on a display with less than 16 characters.

Version:

1.01

Constructor Summary		<i>Page</i>
Label (int charSet, java.lang.String label, int characterFlagField)		375

Method Summary		<i>Page</i>
int getCharacterFlagField ()	Returns the character flag field	377
int getCharset ()	Returns the charset (cf.	375

java.lang.String	getLabel()	404
	Returns the content of the label	

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Detail

Label

```
public Label(int charSet,
            java.lang.String label,
            int characterFlagField)
```

Method Detail

getCharset

```
public int getCharset()
```

Returns the charset (cf. DABConstants.charset*)

getCharacterFlagField

```
public int getCharacterFlagField()
```

Returns the character flag field

getLabel

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

Returns the content of the label

Class DLXObject

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+--dab.data.DABObject
```

```
|
```

```
+--dab.data.DLXObject
```

public Class **DLSObject**

extends [DABObject](#)

The DLSObject represents data of the Dynamic Label Service.

Version:

1.02

Constructor Summary	Page
DLSObject ()	377

Method Summary	Page
int getCharacterFlagField () Returns the CharacterFlagField for the DLS	377
int getCharSet () Returns the charSet of the DLS (cf.	377
java.lang.String getDynamicLabelSegment () Returns the DLS converted to Unicode and without control characters Remark : Not all codetables for Unicode may be implemented on the receiver	377
int getEndofHeadlinePosition () Returns the position of the last character belonging to the Headline inside the DLS	377
int[] getPreferedLineBreakPositions () Returns the positions of the last character before a line break suggested by the broadcaster	378
int[] getPreferedWordBreakPositions () Returns the positions of the last character before a word break suggested by the broadcaster	378
byte[] getRawDynamicLabelSegment () Returns an array of bytes containing the DLS as it is.	377
int getSegmentNumber () Returns the SegmentNumber	378
boolean isCommand () Returns the Command Flag	378
boolean isToggle () Returns the Toggle Flag	378

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail**DLSObject**

```
public DLSObject()
```

Method Detail**getRawDynamicLabelSegment**

```
public byte[] getRawDynamicLabelSegment()
```

Returns an array of bytes containing the DLS as it is.

Remark : The CRC check for the DLS must be successfully passed

getDynamicLabelSegment

```
public java.lang.String getDynamicLabelSegment()  
                        throws DABNotAvailableException
```

Returns the DLS converted to Unicode and without control characters

Remark : Not all codetables for Unicode may be implemented on the receiver

Throws:

[DABNotAvailableException](#) - when the information is not available

getCharSet

```
public int getCharSet()  
          throws DABNotAvailableException
```

Returns the charSet of the DLS (cf. DABConstants.charset*)

Throws:

[DABNotAvailableException](#) - when the information is not available

getCharacterFlagField

```
public int getCharacterFlagField()  
          throws DABNotAvailableException
```

Returns the CharacterFlagField for the DLS

Throws:

[DABNotAvailableException](#) - when the information is not available

getEndofHeadlinePosition

```
public int getEndofHeadlinePosition()  
          throws DABNotAvailableException
```

Returns the position of the last character belonging to the Headline inside the DLS

Throws:

[DABNotAvailableException](#) - when the information is not available

getPreferredLineBreakPositions

```
public int[] getPreferredLineBreakPositions()  
           throws DABNotAvailableException
```

Returns the positions of the last character before a line break suggested by the broadcaster

Throws:

[DABNotAvailableException](#) - when the information is not available

getPreferredWordBreakPositions

```
public int[] getPreferredWordBreakPositions()  
           throws DABNotAvailableException
```

Returns the positions of the last character before a word break suggested by the broadcaster

Throws:

[DABNotAvailableException](#) - when the information is not available

getSegmentNumber

```
public int getSegmentNumber()  
         throws DABNotAvailableException
```

Returns the SegmentNumber

Throws:

[DABNotAvailableException](#) - when the information is not available

isToggle

```
public boolean isToggle()  
           throws DABNotAvailableException
```

Returns the Toggle Flag

Throws:

[DABNotAvailableException](#) - when the information is not available

isCommand

```
public boolean isCommand()  
           throws DABNotAvailableException
```

Returns the Command Flag

Throws:

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

[DABNotAvailableException](#) - when the information is not available

Class *DABObject*

[dab.data](#)

java.lang.Object

|

+--[dab.data.DABObject](#)

Direct Known Subclasses:

[DLSObject](#), [MOTOject](#)

abstract public Class **DABObject**

extends java.lang.Object

The *DABObject* class represents all kind of data that is transported via DAB.

Version:

1.01

Constructor Summary	Page
DABObject()	379

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

DABObject

public *DABObject*()

Class *BWSObject*

[dab.data](#)

java.lang.Object

|

+--[dab.data.DABObject](#)

|

+--[dab.data.MOTOject](#)

|

+--[dab.data.BWSObject](#)

All Implemented Interfaces:

[MOTOjectHeader](#)

public Class **BWSObject**

extends [MOTOject](#)

The BWSObject class represents data that is part of the BWS service

Version:

1.01

Constructor Summary	Page
BWSObject ()	381

Method Summary	Page
java.lang.String getAdditionalHeader ()	381
Returns the additional header (the HTTP header field)	
int getCryptoAlgorithm ()	381
Returns the crypto algorithm for the object	
byte[] getProfileSubset ()	381
Returns the list of profiles for which the object is relevant	
int getScramblingMode ()	381
Returns the scrambling mode for the object	
dab.data.SubscriberInfo getSubscriberInfo ()	381
Returns information about how to subscribe to the service	

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class dab.data. MOTOject
getBody , getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContentType , getCreationTime , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getValidity , getVersionNumber , toString

Methods inherited from class java.lang.Object
equals , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Methods inherited from interface dab.data. MOTOjectHeader
getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContentType , getCreationTime , getExpiration , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getUniqueBodyVersion , getValidity , getVersionNumber

Constructor Detail

BWSObject

```
public BWSObject()
```

Method Detail

getAdditionalHeader

```
public java.lang.String getAdditionalHeader()  
    throws DABNotAvailableException
```

Returns the additional header (the HTTP header field)

Throws:

[DABNotAvailableException](#) - when the content description is not available

getProfileSubset

```
public byte[] getProfileSubset()  
    throws DABNotAvailableException
```

Returns the list of profiles for which the object is relevant

Throws:

[DABNotAvailableException](#) - when the content description is not available

getCryptoAlgorithm

```
public int getCryptoAlgorithm()
```

Returns the crypto algorithm for the object

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

getScramblingMode

```
public int getScramblingMode()
```

Returns the scrambling mode for the object

getSubscriberInfo

```
public dab.data.SubscriberInfo getSubscriberInfo()  
    throws DABNotAvailableException
```

Returns information about how to subscribe to the service

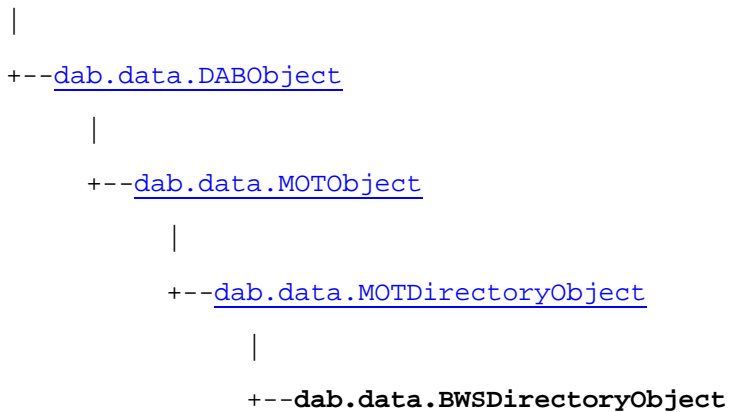
Throws:

[DABNotAvailableException](#) - when the content description is not available

Class *BWSDirectoryObject*

[dab.data](#)

java.lang.Object



All Implemented Interfaces:

[MOTObjectHeader](#)

public Class **BWSDirectoryObject**

extends [MOTDirectoryObject](#)

The *BWSDirectoryObject* class represents the carousel directory of a BWS user application.

Version:

1.01

Constructor Summary		Page
BWSDirectoryObject ()		383

Method Summary		Page
dab.data.BWSDirectoryIndex[]	getDirectoryIndex () returns a list of profile index pages	383

Methods inherited from class dab.data. MOTDirectoryObject
getCarouselPeriod , getContents , getNumberOfObjects

Methods inherited from class dab.data. MOTObject
getBody , getCompressionType , getContentDescription , getContentDescriptionCharset , getContentName , getContentNameCharset , getContentSubtype , getContenttype , getCreationTime , getExpireTime , getLabel , getMimeType , getPriority , getRepetitionDistance , getStartValidity , getTriggerTime , getValidity , getVersionNumber , toString

Generate by DocFlex for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface dab.data.[MOTObjectHeader](#)

[getCompressionType](#), [getContentDescription](#), [getContentDescriptionCharset](#), [getContentName](#), [getContentNameCharset](#), [getContentSubtype](#), [getContenttype](#), [getCreationTime](#), [getExpiration](#), [getExpireTime](#), [getLabel](#), [getMimeType](#), [getPriority](#), [getRepetitionDistance](#), [getStartValidity](#), [getTriggerTime](#), [getUniqueBodyVersion](#), [getValidity](#), [getVersionNumber](#)

Constructor Detail**BWSDirectoryObject**

```
public BWSDirectoryObject()
```

Method Detail**getDirectoryIndex**

```
public dab.data.BWSDirectoryIndex[] getDirectoryIndex()
```

returns a list of profile index pages

Class BWSDirectoryIndex

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+-dab.data.BWSDirectoryIndex
```

```
public Class BWSDirectoryIndex
```

```
extends java.lang.Object
```

The `BWSDirectoryIndex` class represents profile information in a BWS directory

Version:

1.01

Constructor Summary

[BWSDirectoryIndex](#)()

Page

384

Method Summary		Page
java.lang.String	getIndexName() returns the index page name	384
int	getProfileId() returns the profile id that this index is for	384

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

BWSDirectoryIndex

```
public BWSDirectoryIndex()
```

Method Detail

getProfileId

```
public int getProfileId()
```

returns the profile id that this index is for

getIndexName

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

returns the index page name

Class AnnouncementSupport

[dab.data](#)

```
java.lang.Object
```

```
|
```

```
+-dab.data.AnnouncementSupport
```

```
public Class AnnouncementSupport
```

```
extends java.lang.Object
```

AnnouncementSupport represents supported announcement types of a certain DAB service, e.g. News, Traffic and so on.

Version:

1.01

Constructor Summary		Page
	AnnouncementSupport (int announcementSupportFlags)	385

Method Summary		Page
boolean	equals (int announcementSupportFlags) Returns true when this object supports all the given flags; otherwise false	385
boolean	support (int announcement) Returns true when the announcement is supported; otherwise false	385

Methods inherited from class java.lang.Object
<code>equals</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>

Constructor Detail

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

AnnouncementSupport

```
public AnnouncementSupport(int announcementSupportFlags)
```

Method Detail

equals

```
public boolean equals(int announcementSupportFlags)
```

Returns true when this object supports all the given flags; otherwise false

support

```
public boolean support(int announcement)
```

Returns true when the announcement is supported; otherwise false

Class *DABInvalidURLException*

[dab.si](#)

java.lang.Object

|

+--java.lang.Throwable

|

+--java.lang.Exception

|

+--[dab.DABException](#)

|

+--**dab.si.DABInvalidURLException**

All Implemented Interfaces:

java.io.Serializable

public Class **DABInvalidURLException**

extends [DABException](#)

The *DABInvalidURLException* class is used to indicate a invalid DAB URL.

Version:

0.01

Constructor Summary	Page
DABInvalidURLException()	386

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

DABInvalidURLException

public **DABInvalidURLException**()

Class **DABURL**

[dab.si](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

java.lang.Object

|

+--**dab.si.DABURL**

public Class **DABURL**

extends java.lang.Object

Constructor Summary	Page
DABURL (dab.si.ComponentId id) Constructs a DABURL object from the given component <i>id</i> .	388
DABURL (dab.si.EnsembleId id) Constructs a DABURL object from the given ensemble <i>id</i> .	388
DABURL (dab.si.ServiceId id) Constructs a DABURL object from the given service <i>id</i> .	388
DABURL (java.lang.String daburl) Constructs a DABURL object from the given string.	388

Method Summary	Page
dab.si.ComponentId getComponentId () returns the component id of the DAB URL	452
dab.si.DABURL getComponentURL () returns the DAB URL of a component	389
dab.si.EnsembleId getEnsembleId () returns the ensemble id of the DAB URL	414
dab.si.DABURL getEnsembleURL () returns the DAB URL representing the ensemble	388
dab.si.ServiceId getServiceId () returns the service id of the DAB URL	389
dab.si.DABURL getServiceURL () returns the service id of the DAB URL	389
java.lang.String toString () returns DAB URL as string	389

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Detail**DABURL**

```
public DABURL(java.lang.String daburl)
    throws DABInvalidURLException
```

Constructs a DABURL object from the given string.

Throws:

[DABInvalidURLException](#) - if *daburl* represents an invalid DAB URL

DABURL

```
public DABURL(dab.si.EnsembleId id)
    throws DABInvalidURLException
```

Constructs a DABURL object from the given ensemble *id*.

Throws:

[DABInvalidURLException](#) - if *id* represents an invalid ensemble id

DABURL

```
public DABURL(dab.si.ServiceId id)
    throws DABInvalidURLException
```

Constructs a DABURL object from the given service *id*.

Throws:

[DABInvalidURLException](#) - if *id* represents an invalid service id

DABURL

```
public DABURL(dab.si.ComponentId id)
    throws DABInvalidURLException
```

Constructs a DABURL object from the given component *id*.

Throws:

[DABInvalidURLException](#) - if *id* represents an invalid component id

Method Detail**getEnsembleURL**

```
public dab.si.DABURL getEnsembleURL()
    throws DABNotAvailableException
```

returns the DAB URL representing the ensemble

Throws:

[DABNotAvailableException](#) - no ensemble in DAB URL available

getEnsembleId

```
public dab.si.EnsembleId getEnsembleId()  
    throws DABNotAvailableException
```

returns the ensemble id of the DAB URL

Throws:

[DABNotAvailableException](#) - if no ensemble in DAB URL available

getServiceURL

```
public dab.si.DABURL getServiceURL()  
    throws DABNotAvailableException
```

returns the service id of the DAB URL

Throws:

[DABNotAvailableException](#) - if no service in DAB URL available

getServiceId

```
public dab.si.ServiceId getServiceId()  
    throws DABNotAvailableException
```

returns the service id of the DAB URL

Throws:

[DABNotAvailableException](#) - if no service in DAB URL available

getComponentURL

```
public dab.si.DABURL getComponentURL()  
    throws DABNotAvailableException
```

returns the DAB URL of a component

Throws:

[DABNotAvailableException](#) - if no component in DAB URL available

getComponentId

```
public dab.si.ComponentId getComponentId()  
    throws DABNotAvailableException
```

returns the component id of the DAB URL

Throws:

[DABNotAvailableException](#) - no component in DAB URL available

toString

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

returns DAB URL as string

Overrides:

toString in class java.lang.Object

Class **SIId**

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

dab.si

java.lang.Object

|

+--**dab.si.SIId**

Direct Known Subclasses:

[ComponentId](#), [EnsembleId](#), [ServiceId](#)

abstract public Class **SIId**

extends java.lang.Object

The SIId is an identifier for a DAB Ensemble, a DAB Service or a DAB Service Component. It defines a handle to one of these service elements and is used to start and stop services or to query service information.

The identifier for each entity is globally unique. This means an identifier for a component or service includes information about the service context as for instance two services are considered different even if they have the same (DAB) service identifier.

Version:

1.07

Constructor Summary		Page
siid ()		391

Method Summary		Page
int	compareTo (dab.si.SIId siid) This method compares the object with the given object.	391
int	compareTo (java.lang.Object object) This method compares the object with the given object.	391
java.lang.String	getId () Returns an external representation of the identifier in a textual format.	403

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SIIId

```
public SIIId()
```

Method Detail

compareTo

```
public int compareTo(java.lang.Object object)
```

This method compares the object with the given object. The behavior is the same as it is specified in the `compareTo` method of the `java.lang.Comparable` interface.

compareTo

```
public int compareTo(dab.si.SIIId siid)
```

This method compares the object with the given object. The behavior is the same as it is specified in the `compareTo` method of the `java.lang.Comparable` interface.

getId

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

Returns an external representation of the identifier in a textual format. The returned value can be used to construct a service identifier.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Class **ServiceInfo**

[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+-dab.si.ServiceInfo
```

```
public Class ServiceInfo
```

```
extends java.lang.Object
```

`ServiceInfo` is used to represent a service.

Version:

1.04

Constructor Summary		Page
protected	ServiceInfo (dab.si.ServiceId id, int type, dab.si.EnsembleId parent, dab.si.ComponentId[] componentIds, boolean isLocal, int accessControlSystem, boolean hasLabel, dab.data.Label label, boolean hasLanguage, int language, boolean hasIsPrimary, boolean isPrimary, boolean hasRegionId, int regionId, boolean hasRegionLabel, dab.data.Label regionLabel, boolean hasStaticProgrammeType, dab.data.ProgrammeType staticProgrammeType, boolean hasDynamicProgrammeType, dab.data.ProgrammeType dynamicProgrammeType, boolean hasProgrammeNumber, dab.data.ProgrammeNumber programmeNumber, boolean hasTimeOffset, int timeOffset, boolean hasAnnouncementSupport, dab.data.AnnouncementSupport announcementSupport, boolean hasCountry, int country)	393

Method Summary		Page
int	getAccessControlSystem () Returns the access control system (cf.	403
dab.data.AnnouncementSupport	getAnnouncementSupport () Returns the information about announcement support	396
dab.si.ComponentId[]	getComponentIds () Returns a reference to ids of the components of the service	394
int	getCountry () Returns the country information of the service (cf.	400
dab.data.ProgrammeType	getDynamicProgrammeType () Returns the dynamic programme type	395
dab.si.ServiceId	getId () Returns the id of the service	403
dab.data.Label	getLabel () Returns the label of the service	404
int	getLanguage () Returns the language of the service (cf.	404
dab.si.EnsembleId	getParent () Returns the parent ensemble	394
dab.data.ProgrammeNumber	getProgrammeNumber () Returns the programme number	395
int	getRegionId () Returns the region id of the service	395
dab.data.Label	getRegionLabel () Returns the region label of the service	395
dab.data.ProgrammeType	getStaticProgrammeType () Returns the static programme type	395

int	<u>getTimeOffset</u> () Returns the time offset of the service (with respect to the time of the ensemble).	396
int	<u>getType</u> () Returns the service type (cf.	403
boolean	<u>isLocalService</u> () Indicates whether the service is local or not	394
boolean	<u>isPrimaryComponentLanguage</u> () Indicates, whether the language of the service is the language of the primary component	395

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ServiceInfo

```
protected ServiceInfo(dab.si.ServiceId id,
    int type,
    dab.si.EnsembleId parent,
    dab.si.ComponentId[] componentIds,
    boolean isLocal,
    int accessControlSystem,
    boolean hasLabel,
    dab.data.Label label,
    boolean hasLanguage,
    int language,
    boolean hasIsPrimary,
    boolean isPrimary,
    boolean hasRegionId,
    int regionId,
    boolean hasRegionLabel,
    dab.data.Label regionLabel,
    boolean hasStaticProgrammeType,
    dab.data.ProgrammeType staticProgrammeType,
    boolean hasDynamicProgrammeType,
    dab.data.ProgrammeType dynamicProgrammeType,
    boolean hasProgrammeNumber,
    dab.data.ProgrammeNumber programmeNumber,
    boolean hasTimeOffset,
    int timeOffset,
    boolean hasAnnouncementSupport,
    dab.data.AnnouncementSupport announcementSupport,
    boolean hasCountry,
    int country)
```

Method Detail**getId**

```
public dab.si.ServiceId getId()
```

Returns the id of the service

getType

```
public int getType()
```

Returns the service type (cf. DABConstants.serviceType*)

getParent

```
public dab.si.EnsembleId getParent()
```

Returns the parent ensemble

getComponentIds

```
public dab.si.ComponentId[] getComponentIds()
```

Returns a reference to ids of the components of the service

isLocalService

```
public boolean isLocalService()
```

Indicates whether the service is local or not

getAccessControlSystem

```
public int getAccessControlSystem()
```

Returns the access control system (cf. DABConstants.acs*)

getLabel

```
public dab.data.Label getLabel()  
    throws DABNotAvailableException
```

Returns the label of the service

Throws:

[DABNotAvailableException](#) - when the label is not available

getLanguage

```
public int getLanguage()  
    throws DABNotAvailableException
```

Returns the language of the service (cf. DABConstants.language*)

Throws:

[DABNotAvailableException](#) - when the language is not available

isPrimaryComponentLanguage

```
public boolean isPrimaryComponentLanguage()  
           throws DABNotAvailableException
```

Indicates, whether the language of the service is the language of the primary component

Throws:

[DABNotAvailableException](#) - when the information is not available

getRegionId

```
public int getRegionId()  
           throws DABNotAvailableException
```

Returns the region id of the service

Throws:

[DABNotAvailableException](#) - when the id is not available

getRegionLabel

```
public dab.data.Label getRegionLabel()  
           throws DABNotAvailableException
```

Returns the region label of the service

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Throws:

[DABNotAvailableException](#) - when the label is not available

getStaticProgrammeType

```
public dab.data.ProgrammeType getStaticProgrammeType()  
           throws DABNotAvailableException
```

Returns the static programme type

Throws:

[DABNotAvailableException](#) - when the programme type is not available

getDynamicProgrammeType

```
public dab.data.ProgrammeType getDynamicProgrammeType()  
           throws DABNotAvailableException
```

Returns the dynamic programme type

Throws:

[DABNotAvailableException](#) - when the programme type is not available

getProgrammeNumber

```
public dab.data.ProgrammeNumber getProgrammeNumber()  
           throws DABNotAvailableException
```

Returns the programme number

Throws:

[DABNotAvailableException](#) - when not available

getTimeOffset

```
public int getTimeOffset()
    throws DABNotAvailableException
```

Returns the time offset of the service (with respect to the time of the ensemble). The result is returned in minutes. It ranges from -12 hours to 12 hours.

Throws:

[DABNotAvailableException](#) - when the offset is not available

See Also:

[getDate](#)

getAnnouncementSupport

```
public dab.data.AnnouncementSupport getAnnouncementSupport()
    throws DABNotAvailableException
```

Returns the information about announcement support

Throws:

[DABNotAvailableException](#) - when the announcement support is not available

getCountry

```
public int getCountry()
    throws DABNotAvailableException
```

Returns the country information of the service (cf. DABConstants.country*)

Throws:

[DABNotAvailableException](#) - when the country information is not available

Class **ServiceId**

[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+--dab.si.SIId
```

```
|
```

```
+--dab.si.ServiceId
```

```
public Class ServiceId
```

```
extends SIId
```

The ServiceId is an identifier for a DAB service.

Version:

1.05

Constructor Summary		Page
ServiceId (dab.si.ServiceId id)	Constructs a copy of the given ServiceId object.	397
ServiceId (java.lang.String Id)	Constructs a ServiceId object from the given string.	397

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
boolean sameService (dab.si.ServiceId id)	returns true, if id and the called object belong to the same service; otherwise false is returned.	397

Methods inherited from class dab.si. SIId
compareTo , compareTo , getId

Methods inherited from class java.lang.Object
equals , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructor Detail

ServiceId

```
public ServiceId(java.lang.String Id)
```

Constructs a ServiceId object from the given string.

See Also:

[getId](#)

ServiceId

```
public ServiceId(dab.si.ServiceId id)
```

Constructs a copy of the given ServiceId object.

See Also:

[getId](#)

Method Detail

sameService

```
public boolean sameService(dab.si.ServiceId id)
```

returns true, if id and the called object belong to the same service; otherwise false is returned.

Class EnsembleInfo

[dab.si](#)

java.lang.Object

|

+--**dab.si.EnsembleInfo**

public Class **EnsembleInfo**

extends java.lang.Object

EnsembleInfo represents information about a particular ensemble.

Version:

1.03

Constructor Summary		Page
protected	EnsembleInfo (dab.si.EnsembleId id, dab.si.ServiceId[] serviceIds, int frequency, int transmissionMode, boolean hasDate, java.util.Date date, boolean hasLabel, dab.data.Label label, boolean hasCountry, int country)	399

Method Summary		Page
int	getCountry () Returns country information about the ensemble (cf.	400
java.util.Date	getDate () Returns date and time associated with the ensemble (given as local time)	400
int	getFrequency () Returns the frequency of the ensemble in Hz	399
dab.si.EnsembleId	getId () Returns the id of the ensemble	403
dab.data.Label	getLabel () Returns the label of the ensemble	404
dab.si.ServiceId[]	getServiceIds () Returns a reference to the ids for the services that are contained in the ensemble	399
int	getTransmissionMode () Returns the transmission Mode (cf.	428

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail**EnsembleInfo**

```
protected EnsembleInfo(dab.si.EnsembleId id,
                        dab.si.ServiceId[] serviceIds,
                        int frequency,
                        int transmissionMode,
                        boolean hasDate,
                        java.util.Date date,
                        boolean hasLabel,
                        dab.data.Label label,
                        boolean hasCountry,
                        int country)
```

Method Detail**getServiceIds**

```
public dab.si.ServiceId[] getServiceIds()
```

Returns a reference to the ids for the services that are contained in the ensemble

getId

```
public dab.si.EnsembleId getId()
```

Returns the id of the ensemble

getFrequency

```
public int getFrequency()
```

Returns the frequency of the ensemble in Hz

getTransmissionMode

```
public int getTransmissionMode()
```

Returns the transmission Mode (cf. DABConstants.transmissionMode*)

getLabel

```
public dab.data.Label getLabel()
    throws DABNotAvailableException
```

Returns the label of the ensemble

Throws:

[DABNotAvailableException](#) - if the label is not available

getCountry

```
public int getCountry()
    throws DABNotAvailableException
```

Returns country information about the ensemble (cf. DABConstants.country*)

Throws:

[DABNotAvailableException](#) - if the country information is not available

getDate

```
public java.util.Date getDate()
    throws DABNotAvailableException
```

Returns date and time associated with the ensemble (given as local time)

Throws:

[DABNotAvailableException](#) - if the date is not available

Class EnsembleId

[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+--dab.si.SIID
```

```
|
```

```
+--dab.si.EnsembleId
```

```
public Class EnsembleId
```

```
extends SIID
```

The EnsembleId is an identifier for a DAB ensemble.

Version:

1.03

Constructor Summary	Page
EnsembleId (dab.si.EnsembleId id) Constructs a copy of the given EnsembleId object.	401
EnsembleId (java.lang.String Id) Constructs a EnsembleId object from the given string.	401

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class dab.si.SIID[compareTo](#), [compareTo](#), [getId](#)**Methods inherited from class java.lang.Object**[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)**Constructor Detail****EnsembleId**

```
public EnsembleId(java.lang.String Id)
```

Constructs a EnsembleId object from the given string.

See Also:[getId](#)**EnsembleId**

```
public EnsembleId(dab.si.EnsembleId id)
```

Constructs a copy of the given EnsembleId object.

Class ComponentInfo[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+--dab.si.ComponentInfo
```

```
public Class ComponentInfo
```

```
extends java.lang.Object
```

ComponentInfo is used to represent components.

Version:

1.06

Constructor Summary

		Page
protected	ComponentInfo (dab.si.ComponentId id, int type, byte[] data, boolean isPrimary, dab.si.ServiceId[] parentIds, int accessControlSystem, boolean hasLabel, dab.data.Label label, boolean hasLanguage, int language, boolean hasStartObjectId, dab.data.ObjectId startObjectId, boolean hasObjectDirectoryId, dab.data.ObjectId objectDirectoryId, boolean hasAudioComponent, dab.si.ComponentId audioComponent, boolean hasBitrate, int bitrate)	403

Method Summary		Page
int	getAccessControlSystem () Returns the access control system of the component (cf.	403
dab.si.ComponentId	getAudioComponent () Returns the SIIId of the related audio component.	405
int	getBitrate () Returns the maximum bitrate of the component in bits per second.	404
byte[]	getData () Returns the application specific data of the component (i.e. the user application data).	403
dab.si.ComponentId	getId () Returns the id of the component	403
dab.data.Label	getLabel () Returns the label of the component	404
int	getLanguage () Returns the language information (cf.	404
dab.data.ObjectId	getObjectDirectoryId () Returns the id of the object directory	404
dab.si.ServiceId[]	getParentIds () Returns a reference to the ids of the parents	404
dab.data.ObjectId	getStartObjectId () Returns the id of the start object	404
int	getType () Returns the type of the component.	403
boolean	isPrimary () Indicates whether the component is primary or not	403

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ComponentInfo

```
protected ComponentInfo(dab.si.ComponentId id,
                          int type,
                          byte[] data,
                          boolean isPrimary,
                          dab.si.ServiceId[] parentIds,
                          int accessControlSystem,
                          boolean hasLabel,
                          dab.data.Label label,
                          boolean hasLanguage,
                          int language,
                          boolean hasStartObjectId,
                          dab.data.ObjectId startObjectId,
                          boolean hasObjectDirectoryId,
                          dab.data.ObjectId objectDirectoryId,
                          boolean hasAudioComponent,
                          dab.si.ComponentId audioComponent,
                          boolean hasBitrate,
                          int bitrate)
```

Method Detail

getId

```
public dab.si.ComponentId getId()
```

Returns the id of the component

getType

```
public int getType()
```

Returns the type of the component. This is essentially the user application type (cf. DABConstants.componentType*)

getData

```
public byte[] getData()
```

Returns the application specific data of the component (i.e. the user application data).

isPrimary

```
public boolean isPrimary()
```

Indicates whether the component is primary or not

getAccessControlSystem

```
public int getAccessControlSystem()
```

Returns the access control system of the component (cf. DABConstants.acs*)

getParentIds

```
public dab.si.ServiceId[] getParentIds()
```

Returns a reference to the ids of the parents

getLabel

```
public dab.data.Label getLabel()  
    throws DABNotAvailableException
```

Returns the label of the component

Throws:

[DABNotAvailableException](#) - when the label is not available

getLanguage

```
public int getLanguage()  
    throws DABNotAvailableException
```

Returns the language information (cf. DABConstants.language*)

Throws:

[DABNotAvailableException](#) - when the information is not available

getStartObjectId

```
public dab.data.ObjectId getStartObjectId()  
    throws DABNotAvailableException
```

Returns the id of the start object

Throws:

[DABNotAvailableException](#) - when the start object is not available

getObjectDirectoryId

```
public dab.data.ObjectId getObjectDirectoryId()  
    throws DABNotAvailableException
```

Returns the id of the object directory

Throws:

[DABNotAvailableException](#) - when the object directory is not available

getBitrate

```
public int getBitrate()  
    throws DABNotAvailableException
```

Returns the maximum bitrate of the component in bits per second.

Throws:

[DABNotAvailableException](#) - when the bitrate is not available

getAudioComponent

```
public dab.si.ComponentId getAudioComponent()
    throws DABNotAvailableException
```

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

Returns the SIId of the related audio component. Note, that the object has to be a PAD component; otherwise null is returned.

Throws:

[DABNotAvailableException](#) - when not available

Class ComponentId

[dab.si](#)

```
java.lang.Object
```

```
|
```

```
+--dab.si.SIId
```

```
|
```

```
+--dab.si.ComponentId
```

```
public Class ComponentId
```

```
extends SIId
```

The ComponentId is an identifier for a DAB component.

Version:

1.03

Constructor Summary	<i>Page</i>
ComponentId (dab.si.ComponentId id) Constructs a copy of the given ComponentId object.	406
ComponentId (java.lang.String Id) Constructs a ComponentId object from the given string.	406

Methods inherited from class dab.si.[SIId](#)

[compareTo](#), [compareTo](#), [getId](#)

Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

ComponentId

```
public ComponentId(java.lang.String Id)
```

Constructs a ComponentId object from the given string.

See Also:

[getId](#)

ComponentId

```
public ComponentId(dab.si.ComponentId id)
```

Constructs a copy of the given ComponentId object.

Class TuneCnfEvent

[dab.events](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.TuneCnfEvent
```

```
public Class TuneCnfEvent
```

```
extends DABEvent
```

The TuneCnfEvent is generated in response to a tuneReq request.

Version:

1.01

See Also:

[tuneCnf](#)

Constructor Summary		Page
protected	TuneCnfEvent (dab.DABSource source, int result, int tuneState, int tuneFrequency, int transmissionMode, int synchronizationState) Creates a TuneCnfEvent.	407

Method Summary		Page
int	getResult () Returns the status of the Tune command.	447

int	getSynchronizationState() Returns the current synchronization state of the DAB Receiver.	435
int	getTransmissionMode() Returns the DAB transmission mode the DAB receiver has detected.	428
int	getTuneFrequency() Return the frequency currently in use.	432
int	getTuneState() Returns the current tune state independent from the command result indicated by result.	432

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

TuneCnfEvent

```
protected TuneCnfEvent(dab.DABSource source,
                       int result,
                       int tuneState,
                       int tuneFrequency,
                       int transmissionMode,
                       int synchronizationState)
```

Creates a TuneCnfEvent.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Detail

getResult

```
public int getResult()
```

Returns the status of the Tune command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getTuneState

```
public int getTuneState()
```

Returns the current tune state independent from the command result indicated by result. The following values are supported:

- tuneStateNotTuned: The DAB receiver is not tuned to a known frequency. An error has occurred in this case and the following parameters are undefined.
- tuneStateTuned: The DAB receiver is tuned to a frequency specified by tuneFrequency and the following parameters are defined.

getTuneFrequency

```
public int getTuneFrequency()
```

Return the frequency currently in use.

getTransmissionMode

```
public int getTransmissionMode()
```

Returns the DAB transmission mode the DAB receiver has detected. The following values are supported:

- `DABConstants.transmissionMode1`: The found DAB Ensemble is sent in Transmissionmode 1.
- `DABConstants.transmissionMode2`: The found DAB Ensemble is sent in Transmissionmode 2.
- `DABConstants.transmissionMode3`: The found DAB Ensemble is sent in Transmissionmode 3.
- `DABConstants.transmissionMode4`: The found DAB Ensemble is sent in Transmissionmode 4.
- `DABConstants.transmissionModeUnknown`: The transmission mode is unknown.

getSynchronizationState

```
public int getSynchronizationState()
```

Returns the current synchronization state of the DAB Receiver. The following values are supported:

- `DABConstants.stateNotSynchronized`: The DAB Receiver is not synchronized. This is the lowest level of synchronization.
- `DABConstants.stateDABSignalDetected`: The DAB Receiver has detected a DAB Signal.
- `DABConstants.stateTimeAndFrequencySynchronized`: The DAB Receiver is time and frequency synchronized.
- `DABConstants.stateFICReadable`: The Service Information channel is readable. This is the highest level of synchronization.

Class SystemFailureNtfEvent

[dab.events](#)

```
java.lang.Object
```

```

|
+--dab.events.DABEvent
|
+--dab.events.SystemFailureNtfEvent

```

```
public Class SystemFailureNtfEvent
```

```
extends DABEvent
```

The System-Failure event is generated when a fatal error has happened in the system.

Version:

1.02

Constructor Summary		Page
protected	SystemFailureNtfEvent (dab.DABSource source, int reason)	409

Method Summary		Page
int	getReason ()	452
Returns the reason for the system failure (the codes are compatible with DABConstants.result*)		

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

Constructor Detail

SystemFailureNtfEvent

```
protected SystemFailureNtfEvent(dab.DABSource source,
                                int reason)
```

Method Detail

getReason

```
public int getReason()
```

Returns the reason for the system failure (the codes are compatible with DABConstants.result*)

Class SINtfEvent

[dab.events](#)

```
java.lang.Object
```

```

|
+--dab.events.DABEvent
|
+--dab.events.SINtfEvent
```

```
public Class SINtfEvent
```

```
extends DABEvent
```

The SINtfEvent is generated in response to a selectSIReq request.

Version:

1.04

See Also:

[siNtf](#)

Constructor Summary		Page
protected	SINtfEvent (dab.DABSource source, int notification, int updateFlags, dab.si.SIID serviceInfoId, dab.si.EnsembleInfo ensembleInfo, dab.si.ServiceInfo serviceInfo, dab.si.ComponentInfo componentInfo) Creates a SINtfEvent object.	410

Method Summary		Page
dab.si.ComponentInfo	getComponentInfo () If notification signals a component-related notification of type DABConstants.componentAdded or DABConstants.componentChanged and AutoDelivery has been activated with the subscription, then the value refers to a component information object.	447
dab.si.EnsembleInfo	getEnsembleInfo () If notification signals an ensemble-related notification of type DABConstants.ensembleAdded or DABConstants.ensembleChanged and AutoDelivery has been activated with the subscription, then the returned value refers to an ensemble information object.	446
int	getNotification () Returns the notification type.	411
dab.si.ServiceInfo	getServiceInfo () If notification signals a service-related notification of type DABConstants.serviceAdded or DABConstants.serviceChanged and AutoDelivery has been activated with the subscription, then the returned value refers to a service information object.	442
dab.si.SIID	getServiceInfoId () Returns the instance of the service element (Ensemble, Service, Component) that has changed.	412
int	getUpdateFlags () Returns more detailed information about which part of the Service Directory has changed.	435

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SINtfEvent

```
protected SINtfEvent(dab.DABSource source,
    int notification,
    int updateFlags,
    dab.si.SIID serviceInfoId,
    dab.si.EnsembleInfo ensembleInfo,
    dab.si.ServiceInfo serviceInfo,
    dab.si.ComponentInfo componentInfo)
```

Creates a SINtfEvent object.

Method Detail**getNotification**

```
public int getNotification()
```

Returns the notification type. The following values are supported:

Ensemble-related notification:

- **ensembleAdded**: A new DAB Ensemble is available.
- **ensembleRemoved**: A known DAB Ensemble is no longer available. All dependent services and components are also no longer available.
- **ensembleChanged**: A known DAB Ensemble has changed which means its attributes have changed. This case signals changes to the ensemble itself and not changes in linking to child services. This means if a child service is added or removed this is not indicated by a **ensembleChanged** notification.

Service-related notification:

- **serviceAdded**: A new DAB Service is available.
- **serviceRemoved**: A known DAB Service is no longer available. All dependent components are also no longer available.
- **serviceChanged**: A known DAB Service has changed which means its attributes have changed. This case signals changes to the service itself and not changes in linking to child components. This means if a child component is added or removed this is not indicated by a **DABConstants.serviceChanged** notification.

Component-related notification:

- **componentAdded**: A new DAB Component is available.
- **componentRemoved**: A known DAB Component is no longer available.
- **componentChanged**: A known DAB Component has changed which means its attributes have changed.

getUpdateFlags

```
public int getUpdateFlags()
```

Returns more detailed information about which part of the Service Directory has changed. The value is a flag field and supports the following flags depending on the service element type.

In case of a **DABConstants.ensembleAdded** or **DABConstants.ensembleChanged** notification the following values are defined:

- **DABConstants.updatedLabel**: The Ensemble label has changed.
- **DABConstants.updatedCountry**: The Country information which specifies which area is covered by the Ensemble has changed.

In case of a **DABConstants.serviceAdded** or **DABConstants.serviceChanged** notification the following values are defined:

- **DABConstants.updatedLabel**: The Service label has changed.
- **DABConstants.updatedCountry**: The Country information which specifies which area is covered by the Service has changed.
- **DABConstants.updatedTimeOffset**: The time offset for the specified Service has changed.
- **DABConstants.updatedRegion**: The region has changed.
- **DABConstants.updatedStaticProgrammeType**: The static programme type information of the specified audio service has changed.
- **DABConstants.updatedDynamicProgrammeType**: The static programme type information of the specified audio service has changed.
- **DABConstants.updatedAnnouncement**: The announcement information of the specified audio service has changed.
- **DABConstants.updatedLanguage**: The language information of the specified audio service has changed.

- DABConstants.updatedRegionId: The region identifier has changed.
- DABConstants.updatedRegionLabel: The region label has changed.
- DABConstants.updatedAnnouncementSupport: The announcement support information of the specified audio service has changed.
- DABConstants.updatedProgrammeNumber: The programme number has changed

In case of a DABConstants.componentAdded or DABConstants.componentChanged notification the following values are defined:

- DABConstants.updatedLabel: The component label has changed.
- DABConstants.updatedLanguage: The language information of the specified audio component has changed.
- DABConstants.updatedStartObject: In case of a BroadcastWebSite application carried in the related component this indicates that the start object (homepage) is known.
- DABConstants.updatedObjectDirectory: The MOT object directory has changed
- DABConstants.updatedAudioComponent: The link to the audio component has changed
- DABConstants.updatedBitrate: The bitrate has changed.

getServiceInfoId

```
public dab.si.SIId getServiceInfoId()
```

Returns the instance of the service element (Ensemble, Service, Component) that has changed. It can be used in order to request the related information object with the getEnsembleInfo, getServiceInfo or getComponentInfo command.

getEnsembleInfo

```
public dab.si.EnsembleInfo getEnsembleInfo()
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

If notification signals an ensemble-related notification of type DABConstants.ensembleAdded or DABConstants.ensembleChanged and AutoDelivery has been activated with the subscription, then the returned value refers to an ensemble information object. If AutoDelivery is not activated or this is a service-related or component-related notification then null is returned.

getServiceInfo

```
public dab.si.ServiceInfo getServiceInfo()
```

If notification signals a service-related notification of type DABConstants.serviceAdded or DABConstants.serviceChanged and AutoDelivery has been activated with the subscription, then the returned value refers to a service information object. If AutoDelivery is not activated or this is an ensemble-related or component-related notification then null is returned.

getComponentInfo

```
public dab.si.ComponentInfo getComponentInfo()
```

If notification signals a component-related notification of type DABConstants.componentAdded or DABConstants.componentChanged and AutoDelivery has been activated with the subscription, then the value refers to a component information object. If AutoDelivery is not activated or this is an ensemble-related or service-related notification then null is returned.

Class *ServiceFollowingNtfEvent*

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.ServiceFollowingNtfEvent

```

public Class **ServiceFollowingNtfEvent**

extends [DABEvent](#)

The *ServiceFollowingNtfEvent* is generated when a service following action is taken by the receiver.

Version:

1.02

See Also:

[serviceFollowingNtf](#)

Constructor Summary		Page
protected	ServiceFollowingNtfEvent (dab.DABSource source, int action, dab.si.EnsembleId ensembleId, dab.si.ComponentId componentId) Creates an <i>ServiceFollowingNtfEvent</i> event.	414

Method Summary		Page
int	getAction () Returns the service following action code.	414
dab.si.ComponentId	getComponentId () Returns the service identifier of the component that is involved in the action	452
dab.si.EnsembleId	getEnsembleId () Returns the service identifier of the ensemble that is involved in the action	414

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Detail

ServiceFollowingNtfEvent

```
protected ServiceFollowingNtfEvent(dab.DABSource source,  
                                     int action,  
                                     dab.si.EnsembleId ensembleId,  
                                     dab.si.ComponentId componentId)
```

Creates an ServiceFollowingNtfEvent event.

Method Detail

getAction

```
public int getAction()
```

Returns the service following action code. The following codes are possible:

- DABConstants.ServiceFollowingLeavingService: The service following has started. The current ensemble and service are left.
- DABConstants.ServiceFollowingTryingAlternativeService: An alternative frequency is tried.
- DABConstants.ServiceFollowingSelectingService: The service following is finished. The receiver is tuned to a new ensemble and audio service.

getEnsembleId

```
public dab.si.EnsembleId getEnsembleId()
```

Returns the service identifier of the ensemble that is involved in the action

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the service identifier of the component that is involved in the action

Class SelectSICnfEvent

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.SelectSICnfEvent
```

```
public Class SelectSICnfEvent
```

```
extends DABEvent
```

The SelectSICnfEvent is generated in response to a selectSIRReq request.

Version:

1.02

See Also:[selectSICnf](#)

Constructor Summary		Page
protected	SelectSICnfEvent (dab.DABSource source, int result, boolean ensembleInfo, boolean serviceInfo, boolean componentInfo, boolean autoDelivery) Creates an SelectSICnfEvent.	415

Method Summary		Page
boolean	getAutoDelivery () Returns auto delivery.	416
boolean	getComponentInfo () Returns component info.	447
boolean	getEnsembleInfo () Returns the ensemble info.	446
int	getResult () Returns the status of the SelectSI command.	447
boolean	getServiceInfo () Returns service info.	442

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

SelectSICnfEvent

```
protected SelectSICnfEvent(dab.DABSource source,
                           int result,
                           boolean ensembleInfo,
                           boolean serviceInfo,
                           boolean componentInfo,
                           boolean autoDelivery)
```

Creates an SelectSICnfEvent.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the SelectSI command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getEnsembleInfo

```
public boolean getEnsembleInfo()
```

Returns the ensemble info. This value specifies if the DAB client is subscribed to ensemble-specific notifications. The following values are supported:

- true: The client is notified about ensembleAdded, ensembleChanged and ensembleRemoved events.
 - false: The client is not notified about ensembleAdded, ensembleChanged and ensembleRemoved events.
-

getServiceInfo

```
public boolean getServiceInfo()
```

Returns service info. This value specifies if the DAB client is subscribed to service-specific notifications. The following values are supported:

- true: The client is notified about serviceAdded, serviceChanged and serviceRemoved events.
 - false: The client is not notified about serviceAdded, serviceChanged and serviceRemoved events.
-

getComponentInfo

```
public boolean getComponentInfo()
```

Returns component info. This value specifies if the client is subscribed to component-specific notifications. The following values are supported:

- true: The client is notified about componentAdded, componentChanged and componentRemoved events.
 - false: The client is not notified about componentAdded, componentChanged and componentRemoved events.
-

getAutoDelivery

```
public boolean getAutoDelivery()
```

Returns auto delivery. This value specifies if the information related to the notification is sent together with the notification (serviceInfoNtf) or not. The following values are supported:

- true: The serviceInfoNtf method delivers the notification together with the information object. This is only possible for -Added and -Changed notifications but not for -Removed because in the latter case the service element is no longer existing.
- false: The serviceInfoNtf method delivers only the notification. The information object (EnsembleInfo, ServiceInfo or ComponentInfo) itself can be obtained by use of getEnsembleInfoReq, getServiceInfoReq or getComponentInfoReq.

Class *SelectReceptionInfoCnfEvent*

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.SelectReceptionInfoCnfEvent

```

public Class **SelectReceptionInfoCnfEvent**

extends [DABEvent](#)

The *SelectReceptionInfoCnfEvent* is generated in response to a *selectReceptionInfoReq* request.

Version:

1.01

See Also:

[selectReceptionInfoReq](#)

Constructor Summary		Page
protected	SelectReceptionInfoCnfEvent (dab.DABSource source, int result, boolean synchronizationNotifications, boolean bitErrorRateNotifications, boolean muteStateNotifications) Creates a <i>SelectReceptionInfoCnfEvent</i> object.	418

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
boolean	getBitErrorRateNotifications () Returns bit error rate notifications.	418
boolean	getMuteStateNotifications () Returns mute state notifications.	418
int	getResult () Returns the status of the <i>SelectReceptionInfo</i> command.	447
boolean	getSynchronizationNotifications () Returns synchronization notifications.	418

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

SelectReceptionInfoCnfEvent

```
protected SelectReceptionInfoCnfEvent(dab.DABSource source,  
                                       int result,  
                                       boolean synchronizationNotifications,  
                                       boolean bitErrorRateNotifications,  
                                       boolean muteStateNotifications)
```

Creates a SelectReceptionInfoCnfEvent object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the SelectReceptionInfo command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getSynchronizationNotifications

```
public boolean getSynchronizationNotifications()
```

Returns synchronization notifications. The value specifies if the client is notified about state changes concerning DAB signal synchronization. If the returned value is true notifications are provided, if it is false no notifications are provided.

getBitErrorRateNotifications

```
public boolean getBitErrorRateNotifications()
```

Returns bit error rate notifications. The value specifies if the client is notified about state changes concerning the biterrorate. If the returned is true notifications are provided, if it is false no notifications are provided.

getMuteStateNotifications

```
public boolean getMuteStateNotifications()
```

Returns mute state notifications. This value specifies if the client is notified about state changes concerning the mute state of the audio decoder. If the returned value is true notifications are provided, if it is false no notifications are provided.

Class *SelectObjectCnfEvent*

[dab.events](#)

java.lang.Object

|

+--[dab.events.DABEvent](#)

|

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

+--[dab.events.SelectObjectCnfEvent](#)

public Class **SelectObjectCnfEvent**

extends [DABEvent](#)

The *SelectObjectCnfEvent* is generated in response to a *selectObjectReq* request.

Version:

1.02

See Also:

[selectObjectReq](#)

Constructor Summary		Page
protected	SelectObjectCnfEvent (dab.DABSource source, int result, dab.si.ComponentId componentId, dab.data.ObjectId objectId, int requestMode, boolean replaceSelections, java.util.Date accessTime) Creates a <i>SelectObjectCnfEvent</i> object.	420

Method Summary		Page
java.util.Date	getAccessTime () Returns the expected relative access time for delivery of the object.	421
dab.si.ComponentId	getComponentId () Returns the component the object is belonging to.	452
dab.data.ObjectId	getObjectId () Returns the selected object	439
boolean	getReplaceSelections () Returns all current object selections belonging to the component identified by serviceInfoId are replaced with this selection.	421
int	getRequestMode () Returns the current selection mode for the specified object.	420
int	getResult () Returns the status of the <i>SelectObject</i> command.	447

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail**SelectObjectCnfEvent**

```
protected SelectObjectCnfEvent(dab.DABSource source,
                                int result,
                                dab.si.ComponentId componentId,
                                dab.data.ObjectId objectId,
                                int requestMode,
                                boolean replaceSelections,
                                java.util.Date accessTime)
```

Creates a `SelectObjectCnfEvent` object.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Detail**getResult**

```
public int getResult()
```

Returns the status of the `SelectObject` command. If it is equal to `DABConstants.resultOK`, the command was successful. Otherwise an error has occurred.

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the component the object is belonging to.

getObjectId

```
public dab.data.ObjectId getObjectId()
```

Returns the selected object

getRequestMode

```
public int getRequestMode()
```

Returns the current selection mode for the specified object.

- `DABConstants.requestModeOff`: The object selection is removed.
- `DABConstants.requestModeOnce`: The object is requested for one-time delivery. After the first reception from the broadcast channel the object is delivered to the connected DAB client. The client is not notified about new versions.
- `DABConstants.requestModeUpdate`: The object is requested for update delivery. After the first reception from the broadcast channel the object is delivered to the connected client. Additionally each new version of the object is delivered.

getReplaceSelections

```
public boolean getReplaceSelections()
```

Returns all current object selections belonging to the component identified by serviceInfoId are replaced with this selection. If the returned value is true, then all selections are removed. If the returned value is false, then existing selections remain unchanged.

getAccessTime

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

Returns the expected relative access time for delivery of the object.

Class *SelectComponentStreamCnfEvent*

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.SelectComponentStreamCnfEvent
```

```
public Class SelectComponentStreamCnfEvent
```

```
extends DABEvent
```

SelectComponentStreamCnfEvent is generated in response to a selectComponentStreamReq.

Version:

1.02

Constructor Summary		Page
protected	SelectComponentStreamCnfEvent (dab.DABSource source, int result, int streamType, java.io.InputStream stream)	422

Method Summary		Page
int	getResult () Returns the status of the SelectComponentStream command.	447
java.io.InputStream	getStream () Returns the stream.	422
int	getStreamType () Returns the type of the stream (cf.	422

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

`SelectComponentStreamCnfEvent`

```
protected SelectComponentStreamCnfEvent(dab.DABSource source,
                                           int result,
                                           int streamType,
                                           java.io.InputStream stream)
```

Method Detail

`getResult`

```
public int getResult()
```

Returns the status of the `SelectComponentStream` command. If it is equal to `DABConstants.resultOK`, the command was successful. Otherwise an error has occurred.

`getStreamType`

```
public int getStreamType()
```

Returns the type of the stream (cf. `DABConstants.streamType*`).

`getStream`

```
public java.io.InputStream getStream()
```

Returns the stream.

Class `SelectComponentCnfEvent`

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.SelectComponentCnfEvent
```

```
public Class SelectComponentCnfEvent
```

```
extends DABEvent
```

The *SelectComponentCnfEvent* is generated in response to a *selectComponentReq* request.

Version:

1.02

See Also:

[selectComponentCnf](#)

Constructor Summary		Page
protected	SelectComponentCnfEvent (dab.DABSource source, int result, dab.si.ComponentId componentId, int selectionMode) Create a <i>SelectComponentCnfEvent</i> object.	423

Method Summary		Page
dab.si.ComponentId	getComponentId () Returns the component which delivers the started or stopped application.	452
int	getResult () Returns the status of the <i>SelectComponent</i> command.	447
int	getSelectionMode () Return the selection mode for the component.	452

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

SelectComponentCnfEvent

```
protected SelectComponentCnfEvent(dab.DABSource source,
                                   int result,
                                   dab.si.ComponentId componentId,
                                   int selectionMode)
```

Create a *SelectComponentCnfEvent* object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the *SelectComponent* command. If it is equal to *DABConstants.resultOK*, the command was successful. Otherwise an error has occurred.

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the component which delivers the started or stopped application.

getSelectionMode

```
public int getSelectionMode()
```

Return the selection mode for the component. The following flags are supported:

- `DABConstants.selectionModeReplace`: All former selected components of the same type are stopped and the specified component is started. The same type means an audio component replaces any other selected audio component, a data component replaces all other selected independent data components and a programme-associated data component replaces all other selected programme-associated data components.
- `DABConstants.selectionModeAdd`: The application delivered by the specified component is started. Other selected components are not affected.
- `DABConstants.selectionModeRemove`: The selection of the specified component is removed.
- `DABConstants.selectionModeRemoveAll`: All existing component selections are removed. The parameter `serviceId` is set to null in this case.

Class *SelectApplicationCnfEvent*

[dab.events](#)

```
java.lang.Object
```

```

|
+--dab.events.DABEvent
    |
    +--dab.events.SelectApplicationCnfEvent

```

```
public Class SelectApplicationCnfEvent
```

```
extends DABEvent
```

`SelectApplicationCnfEvent` is generated in response to a `selectApplicationReq` request.

Version:

0.2

See Also:

[selectApplicationReq](#)

Constructor Summary		Page
protected	SelectApplicationCnfEvent (<code>dab.DABSource source, int result, dab.DABAppProxy proxy</code>) Creates an <code>SelectApplicationCnfEvent</code> object.	425

Method Summary		Page
dab.DABAppProxy	getApplicationProxy() Returns the proxy for the loaded application.	425
int	getResult() Returns the status of the SelectApplication command.	447

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Constructor Detail

SelectApplicationCnfEvent

```
protected SelectApplicationCnfEvent(dab.DABSource source,
                                     int result,
                                     dab.DABAppProxy proxy)
```

Creates an SelectApplicationCnfEvent object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the SelectApplication command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getApplicationProxy

```
public dab.DABAppProxy getApplicationProxy()
```

Returns the proxy for the loaded application. The value is null, when result != DABConstants.resultOK

Class SearchNtfEvent

[dab.events](#)

java.lang.Object

|

+--[dab.events.DABEvent](#)

|

+--**dab.events.SearchNtfEvent**

public Class **SearchNtfEvent**

extends [DABEvent](#)

SearchNtfEvent is generated in response to a searchReq request.

Version:

1.01

See Also:

[searchNtf](#)

Constructor Summary		Page
protected	SearchNtfEvent (dab.DABSource source, int tuneFrequency, int notifications) Creates a SearchNtfEvent.	426

Method Summary		Page
int	getNotifications () Returns the notification type.	430
int	getTuneFrequency () Returns the currently tuned frequency in Hertz.	432

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Detail

SearchNtfEvent

```
protected SearchNtfEvent(dab.DABSource source,
                          int tuneFrequency,
                          int notifications)
```

Creates a SearchNtfEvent.

Method Detail

getTuneFrequency

```
public int getTuneFrequency()
```

Returns the currently tuned frequency in Hertz.

getNotifications

```
public int getNotifications()
```

Returns the notification type. The value is a flag field supporting the following flags which can be specified together:

- notifications16kHzSteps: A 16 kHz step has been made.
- notificationsTableEntry: A frequency of the specified frequency table has been reached.
- notificationsSearchStarted: Searching for a DAB Ensemble has been started.

Class SearchCnfEvent

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.SearchCnfEvent
```

```
public Class SearchCnfEvent
```

```
extends DABEvent
```

SearchCnfEvent is generated in response to a searchReq request.

Version:

1.01

See Also:

[searchCnf](#)

Constructor Summary		<i>Page</i>
protected	SearchCnfEvent (dab.DABSource source, int result, int tuneState, int tuneFrequency, int transmissionMode, int synchronizationState) Create a SearchCnfEvent	428

Method Summary		<i>Page</i>
int	getResult () Returns the result.	447
int	getSynchronizationState () Returns the synchronization state.	435
int	getTransmissionMode () Returns the transmission mode.	428
int	getTuneFrequency () Return the tune frequency in use.	432

int	getTuneState() Return the tune state.	432
-----	--	-----

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

SearchCnfEvent

```
protected SearchCnfEvent(dab.DABSource source,  
                          int result,  
                          int tuneState,  
                          int tuneFrequency,  
                          int transmissionMode,  
                          int synchronizationState)
```

Create a `SearchCnfEvent`

Method Detail

getResult

```
public int getResult()
```

Returns the result. This value indicates the status of the Tune command. If it is equal to `DABConstants.resultOK`, the command was successful. Otherwise an error has occurred.

getTuneState

```
public int getTuneState()
```

Return the tune state. This value indicates the current tune state independent from the command result indicated by result. The following values are supported:

- `DABConstants.stateNotTuned`: The DAB receiver is not tuned to a known frequency. An error has occurred in this case and the following parameters are undefined.
- `DABConstants.stateTuned`: The DAB receiver is tuned to a frequency specified by `tuneFrequency` and the following parameters are defined.

getTuneFrequency

```
public int getTuneFrequency()
```

Return the tune frequency in use.

getTransmissionMode

```
public int getTransmissionMode()
```

Returns the transmission mode. This value specifies the DAB transmission mode the DAB receiver has detected. The following values are supported:

- DABConstants.transmissionMode1: The found DAB Ensemble is sent in Transmissionmode 1.
- DABConstants.transmissionMode2: The found DAB Ensemble is sent in Transmissionmode 2.
- DABConstants.transmissionMode3: The found DAB Ensemble is sent in Transmissionmode 3.
- DABConstants.transmissionMode4: The found DAB Ensemble is sent in Transmissionmode 4.
- DABConstants.transmissionModeUnknown: The transmission mode is unknown.

getSynchronizationState

```
public int getSynchronizationState()
```

Returns the synchronization state. This value specifies the current synchronization state of the DAB Receiver. The following values are supported:

- DABConstants.stateNotSynchronized: The DAB Receiver is not synchronized. This is the lowest level of synchronization.
- DABConstants.stateDABSignalDetected: The DAB Receiver has detected a DAB Signal.
- DABConstants.stateTimeAndFrequencySynchronized: The DAB Receiver is time and frequency synchronized.
- DABConstants.stateFICReadable: The Service Information channel is readable. This is the highest level of synchronization.

Class ScanNtfEvent

[dab.events](#)

```
java.lang.Object
```

```
|
+--dab.events.DABEvent
|
+--dab.events.ScanNtfEvent
```

```
public Class ScanNtfEvent
```

```
extends DABEvent
```

ScanNtfEvent is generated in response to a scanReq request.

Version:

1.01

See Also:

[scanReq](#)

Constructor Summary		Page
protected	ScanNtfEvent (dab.DABSource source, int tuneFrequency, int notifications) Creates a ScanNtfEvent object.	430

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
int	getNotifications() Returns the notification type.	430
int	getTuneFrequency() Returns the currently tuned frequency in Hertz.	432

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

ScanNtfEvent

```
protected ScanNtfEvent(dab.DABSource source,  
                        int tuneFrequency,  
                        int notifications)
```

Creates a `ScanNtfEvent` object.

Method Detail

getTuneFrequency

```
public int getTuneFrequency()
```

Returns the currently tuned frequency in Hertz.

getNotifications

```
public int getNotifications()
```

Returns the notification type. The returned value is a flag field supporting the following flags which can be specified together:

- `DABConstants.notifications16kHzSteps`: A 16 kHz step has been made.
- `DABConstants.notificationsTableEntry`: A frequency of the specified frequency table has been reached.
- `DABConstants.notificationsEnsembleFound`: A DAB Ensemble has been found.

Class ScanCnfEvent

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.ScanCnfEvent

```

public Class **ScanCnfEvent**

extends [DABEvent](#)

ScanCnfEvent is generated in response to a scanReq request.

Version:

1.01

See Also:

[scanReq](#)

Constructor Summary		Page
protected	ScanCnfEvent (dab.DABSource source, int result, int tuneState, int tuneFrequency, int transmissionModes, int noOfEnsemblesFound) Creates an ScanCnfEvent object.	432

Method Summary		Page
int	getNoOfEnsemblesFound () Returns the number of DAB Ensembles that have been found during the execution of the scan command.	433
int	getResult () Returns the status of the Scan command.	447
int	getTransmissionModes () Returns the transmission modes a DAB Receiver should look for DAB Ensembles.	432
int	getTuneFrequency () Returns the currently tuned frequency.	432
int	getTuneState () Returns the current tune state.	432

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail**ScanCnfEvent**

```
protected ScanCnfEvent(dab.DABSource source,  
                        int result,  
                        int tuneState,  
                        int tuneFrequency,  
                        int transmissionModes,  
                        int noOfEnsemblesFound)
```

Creates an `ScanCnfEvent` object.

Method Detail**getResult**

```
public int getResult()
```

Returns the status of the Scan command. If it is equal to `DABConstants.resultOK`, the command was successful. Otherwise an error has occurred.

getTuneState

```
public int getTuneState()
```

Returns the current tune state. The following values are supported:

- `DABConstants.tuneStateNotTuned`: The DAB Receiver is not tuned to a known frequency.
- `DABConstants.tuneStateTuned`: The DAB Receiver is tuned to a frequency specified by `tuneFrequency`.

getTuneFrequency

```
public int getTuneFrequency()
```

Returns the currently tuned frequency.

getTransmissionModes

```
public int getTransmissionModes()
```

Returns the transmission modes a DAB Receiver should look for DAB Ensembles. The default value is `DABConstants.transmissionModeAutomatic` which means that the receiver is automatically detecting the Transmissionmode. The returned value is a flag field supporting the following flags which can be specified together:

- `DABConstants.transmissionModeAutomatic`: The Transmissionmode is automatically detected
- `DABConstants.transmissionMode1`: At the specified frequency it is tested if a DAB Ensemble is sent in Transmissionmode 1.
- `DABConstants.transmissionMode2`: At the specified frequency it is tested if a DAB Ensemble is sent in Transmissionmode 2.
- `DABConstants.transmissionMode3`: At the specified frequency it is tested if a DAB Ensemble is sent in Transmissionmode 3.

- DABConstants.transmissionMode4: At the specified frequency it is tested if a DAB Ensemble is sent in Transmissionmode 4.

getNoOfEnsemblesFound

```
public int getNoOfEnsemblesFound()
```

Returns the number of DAB Ensembles that have been found during the execution of the scan command.

Class RespondConflictResolutionCnfEvent

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.RespondConflictResolutionCnfEvent
```

```
public Class RespondConflictResolutionCnfEvent
```

```
extends DABEvent
```

Constructor Summary		<i>Page</i>
protected	RespondConflictResolutionCnfEvent (dab.DABSource source, int _result)	433

Method Summary		<i>Page</i>
int	getResult () Returns the status of the respondConflictResolution command.	447

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RespondConflictResolutionCnfEvent

```
protected RespondConflictResolutionCnfEvent(dab.DABSource source,  
                                             int _result)
```

Method Detail

getResult

```
public int getResult()
```

Returns the status of the respondConflictResolution command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

Class ReceptionInfoNtfEvent

[dab.events](#)

```
java.lang.Object
```

```

|
+--dab.events.DABEvent
    |
    +--dab.events.ReceptionInfoNtfEvent

```

```
public Class ReceptionInfoNtfEvent
```

```
extends DABEvent
```

ReceptionInfoNtfEvent is generated in response to a selectReceptionInfoReq request.

Version:

1.01

See Also:

[receptionInfoNtf](#)

Constructor Summary		Page
protected	ReceptionInfoNtfEvent (dab.DABSource source, int updateFlags, int synchronizationState, int bitErrorRateState, int muteState) Creates a ReceptionInfoNtfEvent object.	435

Method Summary		Page
int	getBitErrorRateState () Returns the bit error rate state.	435
int	getMuteState () Returns the mute state.	436
int	getSynchronizationState () Returns the synchronization state.	435
int	getUpdateFlags () Returns the updateFlags.	435

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

ReceptionInfoNtfEvent

```
protected ReceptionInfoNtfEvent(dab.DABSource source,
                                int updateFlags,
                                int synchronizationState,
                                int bitErrorRateState,
                                int muteState)
```

Creates a `ReceptionInfoNtfEvent` object.

Method Detail

getUpdateFlags

```
public int getUpdateFlags()
```

Returns the `updateFlags`. This value is a flag field which indicates if synchronization, biterrrorate and/or mute state has changed. The following values are supported:

- `DABConstants.syncUpdateSynchronizationState`: The synchronization state has changed. The new state is specified by `synchronizationState`.
- `DABConstants.syncUpdateBitErrorRateState`: The biterrrorate state has changed. The new state is specified by `bitErrorRateState`.
- `DABConstants.syncUpdateMuteState`: The mute state has changed. The new state is specified by `muteState`.

getSynchronizationState

```
public int getSynchronizationState()
```

Returns the synchronization state. This value specifies the current synchronization state of the DAB Receiver. The following values are supported:

- `DABConstants.stateSynchronizationStateUnknown`: The synchronization state is not known.
- `DABConstants.stateNotSynchronized`: The DAB Receiver is not synchronized. This is the lowest level of synchronization.
- `DABConstants.stateDABSignalDetected`: The DAB Receiver has detected a DAB Signal.
- `DABConstants.stateTimeAndFrequencySynchronized`: The DAB Receiver is time and frequency synchronized.
- `DABConstants.stateFICReadable`: The Service Information channel is readable. This is the highest level of synchronization.

getBitErrorRateState

```
public int getBitErrorRateState()
```

Returns the bit error rate state. This value specifies the current biterrrorate state. The following values are supported:

- `DABConstants.bitErrorRateLevelUnknown`: The current biterrrorate is unknown.
- `DABConstants.bitErrorRateLevel1`: The biterrrorate is smaller than $5e-4$.

- DABConstants.bitErrorRateLevel2: The biterrorate is smaller than 5e-3.
- DABConstants.bitErrorRateLevel3: The biterrorate is smaller than 5e-2.
- DABConstants.bitErrorRateLevel4: The biterrorate is smaller than 1e-1.
- DABConstants.bitErrorRateLevel5: The biterrorate is equal or larger than 1e-1.

getMuteState

```
public int getMuteState()
```

Returns the mute state. This value specifies the current mute state. The following values are supported:

- DABConstants.muteStateUnknown: The current mute state is unknown.
- DABConstants.muteStateMuting: The DAB Receiver is permanently muting.
- DABConstants.muteStatePartialMuting: Some audio frames were muted.
- DABConstants.muteStateNotMuting: No frame was muted.

Class OperationControlCnfEvent

dab.events

```
java.lang.Object
```

```

|
+--dab.events.DABEvent
|
+--dab.events.OperationControlCnfEvent

```

```
public Class OperationControlCnfEvent
```

```
extends DABEvent
```

OperationControlCnfEvent is generated in response to a operationControlReq request.

Version:

1.01

See Also:

[operationControlCnf](#)

Constructor Summary		Page
protected	OperationControlCnfEvent (dab.DABSource source, int result, int attribute, java.lang.Object value) Create a OperationControlCnfEvent object.	437

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Summary		Page
int	getAttribute() Returns the attribute of the receiver that was involved (cf.	437
int	getResult() Returns the status of the OperationControl command.	447
java.lang.Object	getValue() Returns a copy of the attribute's value.	437

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

OperationControlCnfEvent

```
protected OperationControlCnfEvent(dab.DABSource source,
                                   int result,
                                   int attribute,
                                   java.lang.Object value)
```

Create a OperationControlCnfEvent object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the OperationControl command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getAttribute

```
public int getAttribute()
```

Returns the attribute of the receiver that was involved (cf. DABConstants.operationControl*)

getValue

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

Returns a copy of the attribute's value. This is either the actual value, when a read request was issued, or the former value when a change request was issued.

- DABConstants.operationControlSetVolume: The former volume of the receiver is returned. It is of type Integer in the range from 0 to 100 (percent).
- DABConstants.operationControlGetVolume: The current volume of the receiver is returned. It is of type Integer in the range from 0 to 100 (percent).
- DABConstants.operationControlSetServiceFollowing: The former state of the service following feature is returned. It is of type Boolean: true indicates that the service following was switched on, false indicates that it was switched off.

- `DABConstants.operationControlGetServiceFollowing`: The current state of the service following feature is returned. It is of type Boolean: true indicates that the service following is switched on, false indicates that it is switched off.
- `DABConstants.operationControlGetServiceFollowingNotifications`: null is returned.
- `DABConstants.operationControlSetDRCMMode`: The former state of the DRCMode following feature is returned. It is of type Boolean: true indicates that the DRCMode was switched on, false indicates that it was switched off.
- `DABConstants.operationControlGetDRCMMode`: The current state of DRC mode is returned. It is of type Boolean: true indicates that the DRC mode is switched on, false indicates that it is switched off.
- `DABConstants.operationControlGetDRCMModeNotifications`: null is returned.

See Also:[operationControlReq](#)**Class ObjectNtfEvent**[dab.events](#)

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

java.lang.Object

|

+--[dab.events.DABEvent](#)

|

+--**dab.events.ObjectNtfEvent**public Class **ObjectNtfEvent**extends [DABEvent](#)

The ObjectNtfEvent is generated in response to a ObjectNtfReq request.

Version:

1.02

See Also:[objectNtf](#)

Constructor Summary		Page
protected	ObjectNtfEvent (dab.DABSource source, dab.si.ComponentId componentId, dab.data.ObjectId objectId, int selectionState, dab.data.DABObject object) Creates an ObjectNtfEvent event.	439

Method Summary		Page
dab.si.ComponentId	getComponentId () Returns the component the object is belonging to.	452

dab.data.DABObject	getObject() Returns a reference to a DAB object.	440
dab.data.ObjectId	getObjectId() Returns the id of the selected object.	439
int	getSelectionState() Returns the current selection state.	439

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

ObjectNtfEvent

```
protected ObjectNtfEvent(dab.DABSource source,
                        dab.si.ComponentId componentId,
                        dab.data.ObjectId objectId,
                        int selectionState,
                        dab.data.DABObject object)
```

Creates an ObjectNtfEvent event.

Method Detail

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the component the object is belonging to.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

getObjectId

```
public dab.data.ObjectId getObjectId()
```

Returns the id of the selected object.

getSelectionState

```
public int getSelectionState()
```

Returns the current selection state. The following values are supported:

- DABConstants.selectionStateOK: This message delivers a selected object to the connected DAB client. The object is available by parameter object.
- DABConstants.selectionStateDelayed: Delivery of the selected object is delayed.
- DABConstants.selectionStateTerminated: Transmission of the selected object is terminated. The object selection is removed.

getObject

```
public dab.data.DABObject getObject()
```

Returns a reference to a DAB object. As the DABObject class is just an abstraction of data objects, you have to check the actual type of the returned object (e.g. instanceof MOTObject) to know which kind of object is delivered.

Class LocationInfoNtfEvent

[dab.events](#)

```
java.lang.Object
```

```

|
+--dab.events.DABEvent
|
+--dab.events.LocationInfoNtfEvent

```

```
public Class LocationInfoNtfEvent
```

```
extends DABEvent
```

The LocationInfoNtfEvent represents notifications related to the GetLocationInfo command.

Version:

1.02

See Also:

[locationInfoNtf](#)

Constructor Summary		Page
protected	LocationInfoNtfEvent (dab.DABSource source, java.util.Date timestamp, int[] regionIds, dab.data.LocationInfo info) Creates a LocationInfoNtfEvent object.	441

Method Summary		Page
dab.data.LocationInfo	getLocationInfo () Returns the location info.	441
int[]	getRegionIds () Returns the list of region identifiers.	441
java.util.Date	getTimestamp () Returns the timestamp	441

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Detail

LocationInfoNtfEvent

```
protected LocationInfoNtfEvent(dab.DABSource source,  
                               java.util.Date timestamp,  
                               int[] regionIds,  
                               dab.data.LocationInfo info)
```

Creates a LocationInfoNtfEvent object.

Method Detail

getTimestamp

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

Returns the timestamp

getRegionIds

```
public int[] getRegionIds()
```

Returns the list of region identifiers. When no region ids are available or are not requested, the result is an empty array.

getLocationInfo

```
public dab.data.LocationInfo getLocationInfo()
```

Returns the location info. When the location info was not requested, the result is null.

Class GetServiceInfoCnfEvent

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.GetServiceInfoCnfEvent
```

```
public Class GetServiceInfoCnfEvent
```

```
extends DABEvent
```

The GetServiceInfoCnfEvent is generated in response to a GetServiceInfoReq request.

Version:

1.01

See Also:[getServiceInfoCnf](#)

Constructor Summary		Page
protected	GetServiceInfoCnfEvent (dab.DABSource source, int result, dab.si.ServiceInfo serviceInfo) Creates a GetServiceInfoCnfEvent object.	442

Method Summary		Page
int	getResult () Returns the status of the GetServiceInfo command.	447
dab.si.ServiceInfo	getServiceInfo () Returns a reference to an object which provides information about the DAB Service.	442

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Detail

GetServiceInfoCnfEvent

```
protected GetServiceInfoCnfEvent(dab.DABSource source,
                                   int result,
                                   dab.si.ServiceInfo serviceInfo)
```

Creates a GetServiceInfoCnfEvent object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the GetServiceInfo command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getServiceInfo

```
public dab.si.ServiceInfo getServiceInfo()
```

Returns a reference to an object which provides information about the DAB Service.

Class *GetLocationInfoCnfEvent*

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.GetLocationInfoCnfEvent

```

public Class **GetLocationInfoCnfEvent**

extends [DABEvent](#)

The *GetLocationInfoCnfEvent* is generated in response to a *getLocationInfoReq* request.

Version:

1.01

See Also:

[getLocationInfoCnf](#)

Constructor Summary		Page
protected	GetLocationInfoCnfEvent (dab.DABSource source, int result, int mode, int deliveredDelta, int deliveredAccuracy) Creates a <i>GetLocationInfoCnfEvent</i> object.	444

Method Summary		Page
int	getDeliveredAccuracy () Returns the delivered accuracy of the <i>GetLocationInfo</i> command.	444
int	getDeliveredDelta () Returns the delivered delta of the <i>GetLocationInfo</i> command.	444
int	getMode () Returns the mode of the <i>GetLocationInfo</i> command.	448
int	getResult () Returns the status of the <i>GetLocationInfo</i> command.	447

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Detail

GetLocationInfoCnfEvent

```
protected GetLocationInfoCnfEvent(dab.DABSource source,  
                                   int result,  
                                   int mode,  
                                   int deliveredDelta,  
                                   int deliveredAccuracy)
```

Creates a GetLocationInfoCnfEvent object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the GetLocationInfo command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getMode

```
public int getMode()
```

Returns the mode of the GetLocationInfo command.

See Also:

[getLocationInfoReq](#)

getDeliveredDelta

```
public int getDeliveredDelta()
```

Returns the delivered delta of the GetLocationInfo command.

See Also:

[getLocationInfoReq](#)

getDeliveredAccuracy

```
public int getDeliveredAccuracy()
```

Returns the delivered accuracy of the GetLocationInfo command.

See Also:

[getLocationInfoReq](#)

Class *GetEnsembleInfoCnfEvent*

[dab.events](#)

java.lang.Object

```

|
+--dab.events.DABEvent
|
+--dab.events.GetEnsembleInfoCnfEvent

```

public Class **GetEnsembleInfoCnfEvent**

extends [DABEvent](#)

The *GetEnsembleInfoCnfEvent* is generated in response to a *GetEnsembleInfoReq* request.

Version:

1.01

See Also:

[getEnsembleInfoCnf](#)

Constructor Summary		Page
protected	GetEnsembleInfoCnfEvent (dab.DABSource source, int result, dab.si.EnsembleInfo ensembleInfo) Creates a <i>GetEnsembleInfoCnfEvent</i> object.	445

Method Summary		Page
dab.si.EnsembleInfo	getEnsembleInfo () Returns a reference to an object which provides information about a DAB Ensemble.	446
int	getResult () Returns the status of the <i>GetEnsembleInfo</i> command.	447

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

GetEnsembleInfoCnfEvent

```
protected GetEnsembleInfoCnfEvent(dab.DABSource source,
                                     int result,
                                     dab.si.EnsembleInfo ensembleInfo)
```

Creates a *GetEnsembleInfoCnfEvent* object.

Method Detail**getResult**

```
public int getResult()
```

Returns the status of the GetEnsembleInfo command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

getEnsembleInfo

```
public dab.si.EnsembleInfo getEnsembleInfo()
```

Returns a reference to an object which provides information about a DAB Ensemble.

Class GetComponentInfoCnfEvent[dab.events](#)

```
java.lang.Object
```

```

|
+---dab.events.DABEvent
    |
    +---dab.events.GetComponentInfoCnfEvent

```

```
public Class GetComponentInfoCnfEvent
```

```
extends DABEvent
```

The GetComponentInfoCnfEvent is generated in response to a GetComponentInfoReq request.

Version:

1.01

See Also:

[getComponentInfoCnf](#)

Constructor Summary		<i>Page</i>
protected	GetComponentInfoCnfEvent (dab.DABSource source, int result, dab.si.ComponentInfo componentInfo) Create a GetComponentInfoCnfEvent object.	447

Method Summary		<i>Page</i>
dab.si.ComponentInfo	getComponentInfo () Returns information about the subscribed DAB Component.	447
int	getResult () Returns the status of the GetComponentInfo command.	447

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

GetComponentInfoCnfEvent

```
protected GetComponentInfoCnfEvent(dab.DABSource source,  
                                   int result,  
                                   dab.si.ComponentInfo componentInfo)
```

Create a GetComponentInfoCnfEvent object.

Method Detail

getResult

```
public int getResult()
```

Returns the status of the GetComponentInfo command. If it is equal to DABConstants.resultOK, the command was successful. Otherwise an error has occurred.

GetComponentInfo

```
public dab.si.ComponentInfo GetComponentInfo()
```

Returns information about the subscribed DAB Component.

Class **DRCModeNtfEvent**

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.DRCModeNtfEvent
```

```
public Class DRCModeNtfEvent
```

```
extends DABEvent
```

The DRCModeNtfEvent is generated when a DRC mode change is taken by the receiver.

Version:

1.02

See Also:[serviceFollowingNtf](#)

Constructor Summary		Page
protected	DRCModeNtfEvent (dab.DABSource source, boolean currentState)	448
Creates an DRCModeNtfEvent event.		

Method Summary		Page
boolean	getMode ()	448
Returns the DRC mode.		

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Generate by DocFlextor for Javadoc 1.0 Evaluation Version

Constructor Detail

DRCModeNtfEvent

```
protected DRCModeNtfEvent(dab.DABSource source,
                           boolean currentState)
```

Creates an DRCModeNtfEvent event.

Method Detail

getMode

```
public boolean getMode()
```

Returns the DRC mode. The return value is true, if it is now switched on; it is false when it is now switched off.

Class DABEvent

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

Direct Known Subclasses:

[ComponentNtfEvent](#), [ConflictResolutionNtfEvent](#), [DRCModeNtfEvent](#), [GetComponentInfoCnfEvent](#), [GetEnsembleInfoCnfEvent](#), [GetLocationInfoCnfEvent](#), [GetServiceInfoCnfEvent](#), [LocationInfoNtfEvent](#), [ObjectNtfEvent](#), [OperationControlCnfEvent](#), [ReceptionInfoNtfEvent](#), [RespondConflictResolutionCnfEvent](#), [ScanCnfEvent](#), [ScanNtfEvent](#), [SearchCnfEvent](#), [SearchNtfEvent](#), [SelectApplicationCnfEvent](#), [SelectComponentCnfEvent](#), [SelectComponentStreamCnfEvent](#), [SelectObjectCnfEvent](#), [SelectReceptionInfoCnfEvent](#), [SelectSICnfEvent](#), [ServiceFollowingNtfEvent](#), [SINtfEvent](#), [SystemFailureNtfEvent](#), [TuneCnfEvent](#)

public Class **DABEvent**

extends java.lang.Object

DABEvent is the superclass for all events used inside the DAB package. Note: to support J2ME MIDP profile DABEvent does not extend java.util.EventObject anymore

Version:

1.02

Constructor Summary		Page
protected	DABEvent (dab.DABSource source)	449

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

DABEvent

protected **DABEvent**(dab.DABSource source)

Class ConflictResolutionNtfEvent

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

[dab.events](#)

java.lang.Object

|

+--[dab.events.DABEvent](#)

|

+--**dab.events.ConflictResolutionNtfEvent**

public Class **ConflictResolutionNtfEvent**

extends [DABEvent](#)

Constructor Summary		Page
protected	ConflictResolutionNtfEvent (dab.DABSource source, int _transaction, int _turn, int _operation, int _suboperation)	450

Method Summary		Page
int	getOperation() Gives back a code of the involved operation (cf.	451
int	getSuboperation() Gives back a code of the involved suboperation (cf.	451
int	getTransaction() Delivers the transaction number.	450
int	getTurn() Returns a code for the turn of the resource conflict resolution protocol: <ul style="list-style-type: none">• <code>DABConstants.conflictResolutionTurnProceed</code>: This is sent to the client which requested the operation.	450

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

ConflictResolutionNtfEvent

```
protected ConflictResolutionNtfEvent(dab.DABSource source,
                                     int _transaction,
                                     int _turn,
                                     int _operation,
                                     int _suboperation)
```

Method Detail

getTransaction

```
public int getTransaction()
```

Delivers the transaction number. This can be used to provide a transaction context.

getTurn

```
public int getTurn()
```

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Returns a code for the turn of the resource conflict resolution protocol:

- `DABConstants.conflictResolutionTurnProceed`: This is sent to the client which requested the operation. It indicates that there is a resource conflict. The client is asked whether he likes to proceed.
- `DABConstants.conflictResolutionTurnProbe`: This notification is sent to all clients in order to probe for their willingness to release the needed resources.
- `conflictResolutionTurnStop`: The client is asked to stop the indicated operation in order to release the resources.
- `DABConstants.conflictResolutionTurnPreempt`: The client is informed that the indicated operation was stopped. This action shall normally only be taken, when the client failed to do a stop in the previous turn.

getOperation

```
public int getOperation()
```

Gives back a code of the involved operation (cf. DABConstants.conflictResolutionOperation*)

getSuboperation

```
public int getSuboperation()
```

Gives back a code of the involved suboperation (cf. DABConstants.conflictResolutionSuboperation*)

Class ComponentNtfEvent

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.DABEvent
```

```
|
```

```
+--dab.events.ComponentNtfEvent
```

```
public Class ComponentNtfEvent
```

```
extends DABEvent
```

Version:

```
1.02
```

Constructor Summary		Page
protected	ComponentNtfEvent (dab.DABSource source, int reason, dab.si.ComponentId componentId, int selectionMode)	452

Method Summary		Page
dab.si.ComponentId	getComponentId () Returns the component which is involved	452
int	getReason () Returns the reason for change of the selectionMode (the code is compatible with DABConstants.result*).	452
int	getSelectionMode () Returns the new selection mode for the component.	452

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Detail**ComponentNtfEvent**

```
protected ComponentNtfEvent(dab.DABSource source,
                             int reason,
                             dab.si.ComponentId componentId,
                             int selectionMode)
```

Method Detail**getReason**

```
public int getReason()
```

Returns the reason for change of the selectionMode (the code is compatible with DABConstants.result*).

getComponentId

```
public dab.si.ComponentId getComponentId()
```

Returns the component which is involved

getSelectionMode

```
public int getSelectionMode()
```

Returns the new selection mode for the component.

See Also:

[getSelectionMode](#)

Class AppStateChangeEvent

[dab.events](#)

```
java.lang.Object
```

```
|
```

```
+--dab.events.AppStateChangeEvent
```

```
public Class AppStateChangeEvent
```

```
extends java.lang.Object
```

AppStateChangeEvent reflects state changes in an application. Note: to support J2ME MIDP profile AppStateChangeEvent does not extend java.util.EventObject anymore

Version:

0.3

See Also:[AppStateChangeListener](#), "Digital Video Broadcasting (DVB) Multimedia Home Platform (MHP), ETSI TS 101 812"

Constructor Summary		Page
protected	AppStateChangeEvent (dab.AppStateChangeEventSource source, int fromState, int toState, boolean failed)	453

Method Summary		Page
int	getFromState () Returns the state from which the application was switching	453
int	getToState () Returns the state to which the application switched	453
boolean	hasFailed () Indicates whether the switching failed (=true) or not (=false)	454

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Constructor Detail

AppStateChangeEvent

```
protected AppStateChangeEvent(dab.AppStateChangeEventSource source,
                               int fromState,
                               int toState,
                               boolean failed)
```

Method Detail

getFromState

```
public int getFromState()
```

Returns the state from which the application was switching

getToState

```
public int getToState()
```

Returns the state to which the application switched

hasFailed

```
public boolean hasFailed()
```

Indicates whether the switching failed (=true) or not (=false)

Class XletStateChangeException

[dab.xlet](#)

```
java.lang.Object
|
+-- java.lang.Throwable
    |
    +-- java.lang.Exception
        |
        +-- dab.xlet.XletStateChangeException
```

All Implemented Interfaces:

java.io.Serializable

```
public Class XletStateChangeException
```

```
extends java.lang.Exception
```

The XletStateChangeException is used to signal errors in the execution of some of the Xlet methods.

Version:

0.2

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Interface XletContext

[dab.xlet](#)

```
public Interface XletContext
```

The XletContext can be used by the Xlet to communicate with its application controller.

Version:
0.2

Method Summary		Page
void	destroyed () This method indicates to the application controller that the Xlet has destroyed itself in the same way as if destroyXlet has been called	455
java.lang.Object	getXletProperty (java.lang.String key) Requests a property from the environment.	455
void	paused () The Xlet indicates that it has paused its activities in the same manner as if pauseXlet has been called.	455
void	resumeRequest () If the Xlet has paused its activities, it may use this method to signal the application controller to start the Xlet again.	455

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

Method Detail

destroyed

```
public void destroyed()
```

This method indicates to the application controller that the Xlet has destroyed itself in the same way as if destroyXlet has been called

See Also:

[destroyXlet](#)

getXletProperty

```
public java.lang.Object getXletProperty( java.lang.String key)
```

Requests a property from the environment. The value of the property with the given key is returned or is NULL if no such property exists.

paused

```
public void paused()
```

The Xlet indicates that it has paused its activities in the same manner as if pauseXlet has been called.

See Also:

[pauseXlet](#)

resumeRequest

```
public void resumeRequest()
```

If the Xlet has paused its activities, it may use this method to signal the application controller to start the Xlet again. Note, it is necessary to use a different thread in the application controller for calling startXlet to avoid deadlocks.

Interface Xlet

[dab.xlet](#)

public Interface **Xlet**

The Xlet is an abstract class that defines the application model used in the dab package.

The `xlet` class contains all the basic methods for controlling and initialising the Xlet component. The method described below are in its essence signals of state changes to the Xlet. This means if the application controller of the Xlet is changing its state, it will call the respective method to indicate the state change to the Xlet.

For a more detailed description of the Xlet model see the java TV API documentation (<http://java.sun.com/products/javatv/>)

Version:

0.2

See Also:

[DABAppProxy](#)

Method Summary		Page
void	destroyXlet (boolean unconditional) The Xlet is requested to stop its operations and to release all the resources, it has allocated.	457
void	initXlet (dab.xlet.XletContext ctx) If this method is called, the Xlet may initialise itself.	456
void	pauseXlet () The Xlet is asked to temporarily freeze its operations.	457
void	startXlet () The Xlet is requested to start its operations.	457

Method Detail

initXlet

```
public void initXlet(dab.xlet.XletContext ctx)
    throws XletStateChangeException
```

If this method is called, the Xlet may initialise itself. The effect is that the Xlet is ready for execution, but has not started its operation, like after a call of `pauseXlet`.

Throws:

[dab.xlet.XletStateChangeException](#) - The exception is thrown if the initialization was not possible.

Generate by DocFlexor for Javadoc 1.0 Evaluation Version

startXlet

```
public void startXlet()  
    throws XletStateChangeException
```

The Xlet is requested to start its operations. In this state the Xlet may hold shared resources. This method will only be called after a call of `initXlet` or `pauseXlet`

Throws:

[dab.xlet.XletStateChangeException](#) - The exception is thrown if the Xlet could not be started.

pauseXlet

```
public void pauseXlet()
```

The Xlet is asked to temporarily freeze its operations. This also means that shared resources are released.

destroyXlet

```
public void destroyXlet(boolean unconditional)  
    throws XletStateChangeException
```

The Xlet is requested to stop its operations and to release all the resources, it has allocated. No other methods are called afterwards, if the `destroyXlet` returns successfully.

Parameters:

`unconditional` - this parameter indicates that any requests of the Xlet not to be destroyed are ignored (if set to true) or not (if set to false).

Throws:

[dab.xlet.XletStateChangeException](#) - The exception is thrown if the Xlet (or its depending resources) could not be destroyed or if it requests not to be destroyed. If the request is accepted, `destroyXlet` will be called again at a later time.

Annex B (informative): Bibliography

Implementing Protection Domains in the Java Development Kit 1.2 (*Li Gong and Roland Schemers*) - Proceeding of Internet Society ...

Java Security Architecture (<http://Java.sun.com/.../security-spec.html>)

Security reference Model for JDK 1.0.2 by M. Erdos, B. Hartman, M. Mueller (13 November 1996) - Sun specification

Java TV API Specification (<http://java.sun.com/products/javatv/>)

DVB Java specification (http://www.dvb.org/dvb_technology/framesets/standspec-fr.html)

Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (GSM 11.14 version 7.2.0 Release 1998)

Connected, Limited Device Configuration (March 8, 2000)
(http://Java.sun.com/aboutJava/communityprocess/jsr/jsr_030_j2melc.html)

PDA Profile for J2ME (http://Java.sun.com/aboutJava/communityprocess/jsr/jsr_075_pda.html)

J2ME Connected Device Configuration (http://Java.sun.com/aboutJava/communityprocess/jsr/jsr_036_j2mecd.html)

Requirements for Runtime Package (TF VM 44 - GNM - 17.2.2000)

JAR Archive documentation (<http://www.Javasoft.com/j2se/1.3/docs/guide/jar/index.html>)

Manifest (<http://www.Javasoft.com/j2se/1.3/docs/guide/jar/jar.html#The META-INF directory>)

The HAVi Specification. (<http://www.havi.org>)

PersonalJava 1.1 <http://Java.sun.com/products/personalJava/>

PersonalJava datasheet (http://Java.sun.com/products/personalJava/pJava_ds.html)

DAB Java: The Runtime Package, WorldDAB TF-VM (Antonio Barletta)

DAB Java User Application Signalling, WorldDAB TF-VM

PersonalJava and J2ME <http://Java.sun.com/products/personalJava/faq.html#A11>

Design Pattern, Element of Reusable Object-oriented Software by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides (Addison-Wesley - ISBN0201633612)

History

Document history		
V1.1.1	March 2002	Publication
V1.2.1	August 2004	Major Bugfixes, OSGI Integration, Signalling and Transport issues added, Clarification on DAB-Java Application Control added

Dateiname: ETSI TS 101 993 V1.2.1_withAnnexA_20040826.1600_nochanges
Verzeichnis: D:\doc\dab\technology\DABJava\taskforce\actualspec\specification
Vorlage: C:\ETSI\TEMPLATES\ETSI 'NEW' DELIVERABLES\ETSIW_80.DOT
Titel: ETSI TS 101 993 V1.1.1
Thema: Digital Audio Broadcasting (DAB)
Autor: SGU
Stichwörter: ETS
Kommentar:
Erstelldatum: 26.08.2004 3:48
Änderung Nummer: 9
Letztes Speicherdatum: 26.08.2004 5:24
Zuletzt gespeichert von: Thomas Hack
Letztes Druckdatum: 26.08.2004 5:24
Nach letztem vollständigen Druck
Anzahl Seiten: 460
Anzahl Wörter: 118.827 (ca.)
Anzahl Zeichen: 748.617 (ca.)