

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Terrestrial digital multimedia broadcasting (T-DMB) receivers –
Part 3: Common API**

**Récepteurs pour diffusion multimédia numérique terrestre (T-DMB) –
Partie 3: API commune**





THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2013 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,...).

It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available on-line and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Liens utiles:

Recherche de publications CEI - www.iec.ch/searchpub

La recherche avancée vous permet de trouver des publications CEI en utilisant différents critères (numéro de référence, texte, comité d'études,...).

Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

Just Published CEI - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications de la CEI. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (VEI) en ligne.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.



IEC 62516-3

Edition 1.0 2013-03

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Terrestrial digital multimedia broadcasting (T-DMB) receivers –
Part 3: Common API**

**Récepteurs pour diffusion multimédia numérique terrestre (T-DMB) –
Partie 3: API commune**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

U

ICS 33.160.25; 33.170

ISBN 978-2-83220-685-0

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	3
1 Scope.....	5
2 Normative references	5
3 Abbreviations	5
4 T-DMB common API overview	6
4.1 T-DMB receiver overview	6
4.2 T-DMB receiver ASIC block	6
4.3 Host processor block	6
4.3.1 General	6
4.3.2 T-DMB driver (hardware abstraction layer) sub-block.....	7
4.3.3 T-DMB ASIC specific software sub-block	7
4.3.4 T-DMB common APIs sub-block	7
4.3.5 T-DMB receiver middleware sub-block.....	7
4.4 Hardware interface block.....	8
5 API description	8
5.1 T-DMB common APIs	8
5.2 Command types	9
5.2.1 General	9
5.2.2 Get receiver capability.....	9
5.2.3 Tuning	10
5.2.4 Searching	11
5.2.5 Scanning	14
5.2.6 Selecting a T-DMB service	16
5.2.7 Selecting a slideshow or a dynamic label service.....	18
5.2.8 Selecting a broadcast website service	19
5.2.9 Get T-DMB service information	21
5.2.10 Monitoring reception qualities	22
Annex A (informative) Examples of the classes used in T-DMB APIs	25
Bibliography.....	28
Figure 1 – Block diagram of a typical T-DMB receiver	6
Figure 2 – Three different command patterns.....	8
Figure 3 – Get receiver capability	10
Figure 4 – Tuning.....	10
Figure 5 – Searching.....	12
Figure 6 – Scanning.....	14
Figure 7 – Selecting a T-DMB service	17
Figure 8 – Selecting a slideshow or a dynamic label service	18
Figure 9 – Selecting a broadcast website service.....	19
Figure 10 – Get T-DMB service information	21
Figure 11 – Monitoring reception qualities.....	23

INTERNATIONAL ELECTROTECHNICAL COMMISSION

TERRESTRIAL DIGITAL MULTIMEDIA BROADCASTING (T-DMB) RECEIVERS –

Part 3: Common API

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62516-3 has been prepared by technical area 1: Terminals for audio, video and data services and contents, of IEC technical committee 100: Audio, video and multimedia systems and equipment.

The text of this standard is based on the following documents:

CDV	Report on voting
100/2020/CDV	100/2110/RVC

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62516 series, published under the general title *Terrestrial digital multimedia broadcasting (T-DMB) receivers*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

TERRESTRIAL DIGITAL MULTIMEDIA BROADCASTING (T-DMB) RECEIVERS –

Part 3: Common API

1 Scope

This part of IEC 62516 describes the T-DMB common application program interface (API). It provides a software platform that, when combined with the T-DMB O/S, forms a universal interface for application programs. This interface allows application programs to be written in such a way that they run on any T-DMB receiver unit, as described in IEC 62516-1:2009 and IEC 62516-2:2011 regardless of its manufacturer.

This part of IEC 62516 also defines a software environment that allows multiple application programs to be interoperable on a single receiver unit by sharing the fixed resources of the receiver, and it provides a set of interfaces that the T-DMB middleware and the ASIC specific software use.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62516-1:2009, *Terrestrial digital multimedia broadcasting (T-DMB) receivers – Part 1: Basic requirements*

IEC 62516-2:2011, *Terrestrial digital multimedia broadcasting (T-DMB) receivers – Part 2: Interactive data services using BIFS*

ETSI EN 300 401 v1.3.3, *Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers*

3 Abbreviations

ADC	Analog to Digital Converter
API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
FIC	Fast Information Channel
HAL	Hardware Abstraction Layer
ISR	Interrupt Service Routine
MAC	Media Access Control
PAD	Program Associated Data
RF	Radio Frequency
R-S	Reed Solomon
SDIO	Secure Digital Input/Output
SI	Service Identifier
T-DMB	Terrestrial-Digital Multimedia Broadcasting

O/S Operating System

4 T-DMB common API overview

4.1 T-DMB receiver overview

A T-DMB receiver provides the device functionality specified in the T-DMB receivers (see IEC 62516-1:2009 and IEC 62516-2:2011). Figure 1 shows the block diagram of a typical T-DMB receiver. For the T-DMB receiver depicted in Figure 1, only those blocks that conform to the scope of this standard are shown.

Figure 1 also shows the T-DMB common API with respect to the T-DMB receiver block diagram.

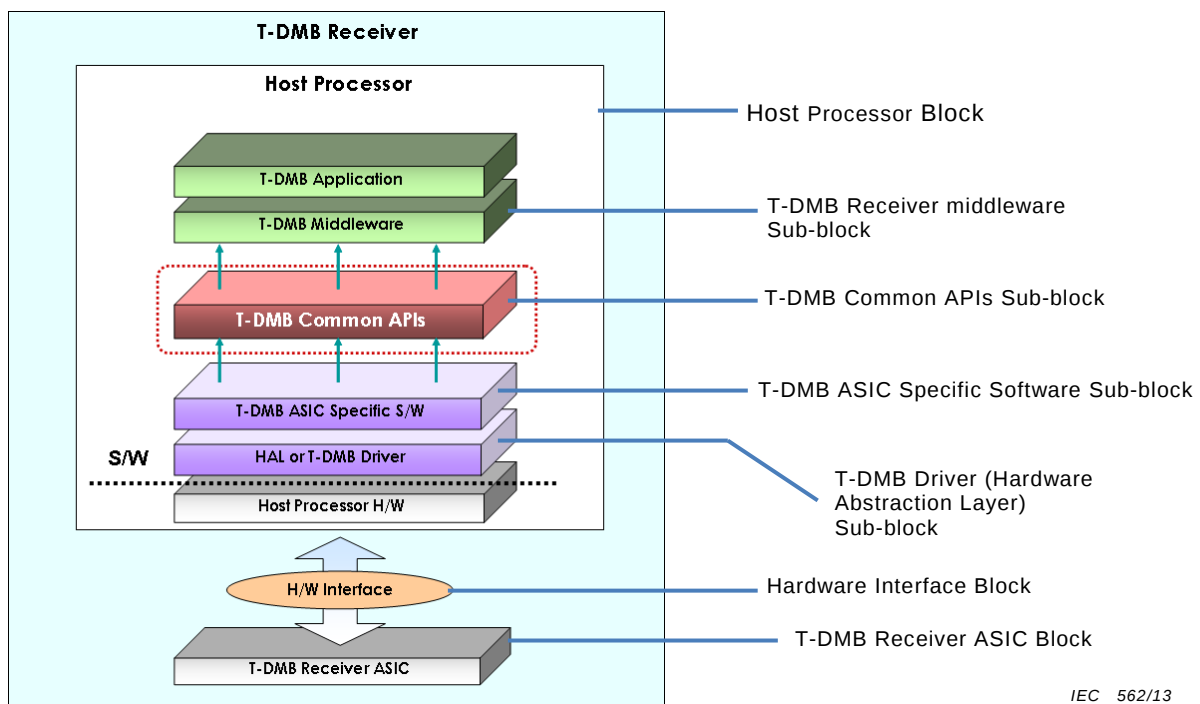


Figure 1 – Block diagram of a typical T-DMB receiver

4.2 T-DMB receiver ASIC block

The T-DMB receiver ASIC block represents the semiconductor hardware that provides the functionality of demodulating a T-DMB signal and retrieving data carried by the T-DMB physical layer. This block provides functionality like RF front-end, ADC, timing and frequency estimation, channel estimation, viterbi decoding, etc. In short this block provides the complete physical layer implementation of ETSI EN 300 401 v1.3.3. Depending upon the implementation, this block can also provide outer decoding functionality (e.g. R-S decoding and/or convolutional de-interleaving).

4.3 Host processor block

4.3.1 General

The host processor block represents the T-DMB functionality provided by the host processor in a T-DMB based device. In other words, this represents the host processor hardware and the software implementation residing in the host processor. The host processor block retrieves and processes the T-DMB information obtained from the T-DMB receiver ASIC block. The T-DMB information retrieved consists of multiplex configuration information received on fast

information channel (FIC), content received on the main service channel. This block communicates with the T-DMB receiver ASIC block to retrieve the information received from the T-DMB signal. The host processor block consists of the following functional sub-blocks.

4.3.2 T-DMB driver (hardware abstraction layer) sub-block

The T-DMB Driver or hardware abstraction layer (HAL) Block represents the driver level software in the main processor that directly interfaces with the T-DMB receiver ASIC block. The T-DMB driver sub-block provides controller functions (e.g. turning on or turning off the T-DMB receiver ASIC block) and data exchange functions (e.g. retrieving the data from the T-DMB receiver ASIC block or conveying the characteristics of a sub-channel to be received) for a given T-DMB receiver ASIC hardware. The T-DMB Driver software is specific to the type of hardware interface mechanism that exists between the Host Processor Block and the T-DMB Receiver ASIC block.

For example, the T-DMB driver software will be different depending upon whether the hardware interface between the main processor and the T-DMB receiver ASIC block is interrupt driven, implemented with memory mapped address/registers or packet based transaction interface like SDIO. Some examples of tasks performed by T-DMB driver sub-block are:

- hardware interactions such as initialization, sleep or wakeup triggers;
- data exchange with hardware such as emptying hardware buffers into main memory or providing ISR implementation.

The T-DMB driver software functions are tightly coupled with the T-DMB receiver ASIC hardware and are considered time sensitive in nature. Therefore the T-DMB driver software is typically given a higher priority with respect to other sub-blocks shown. For example, the T-DMB driver performs the tasks of retrieving the data received by the T-DMB receiver ASIC block or instructing the T-DMB receiver ASIC block to tune to a frequency as requested by the application layer.

4.3.3 T-DMB ASIC specific software sub-block

T-DMB ASIC specific software sub-block provides the MAC layer functionality not covered by the T-DMB driver sub-block. Depending upon the division of MAC layer functionality across different sub-blocks, it may provide complete or partial MAC layer functionality. At the least, the T-DMB ASIC specific software sub-block is expected to provide high level MAC layer functionality that is not practical to be delegated to T-DMB driver sub-block. It interfaces with the T-DMB receiver middleware sub-block using the T-DMB common APIs.

4.3.4 T-DMB common APIs sub-block

The T-DMB common APIs sub-block defines the interfaces that allow the T-DMB ASIC specific software sub-block to communicate with the T-DMB receiver middleware. Any T-DMB receiver middleware that adheres to the interfaces defined by the T-DMB common APIs will work with any T-DMB ASIC specific software sub-block that adheres to these interfaces as well. More details on this interface are provided in the rest of this standard.

4.3.5 T-DMB receiver middleware sub-block

The T-DMB receiver middleware sub-block communicates with the T-DMB ASIC specific software sub-block using the T-DMB common APIs. The T-DMB receiver middleware implements the control and stream layer and provides the interface with application layer protocols. It triggers the T-DMB ASIC specific software to receive the specified contents as requested by the application layer. It acts on the notifications or content provided by the T-DMB ASIC specific software. It delivers any content received from the T-DMB ASIC specific software to the application layer protocols.

4.4 Hardware interface block

The hardware interface block represents the hardware interface mechanism that exists between the host processor block and the T-DMB receiver ASIC block. This interface provides the communication and data exchange functionality. The T-DMB driver sub-block uses this block to exchange commands and data with the T-DMB receiver ASIC block. The hardware interface block can be any desired interface, such as proprietary bus interface or a standard based interface (e.g. SDIO).

5 API description

5.1 T-DMB common APIs

This clause provides a detailed description of each T-DMB common API. The API function prototype details are provided along with defined types needed by the T-DMB common APIs.

Commands are executed by sending requests, confirmations and notifications. Figure 2 shows three different command patterns. These are used in the T-DMB common APIs.

If commands are interleaved which means two commands running at the same time, an arbitrary sequence of message types is possible.

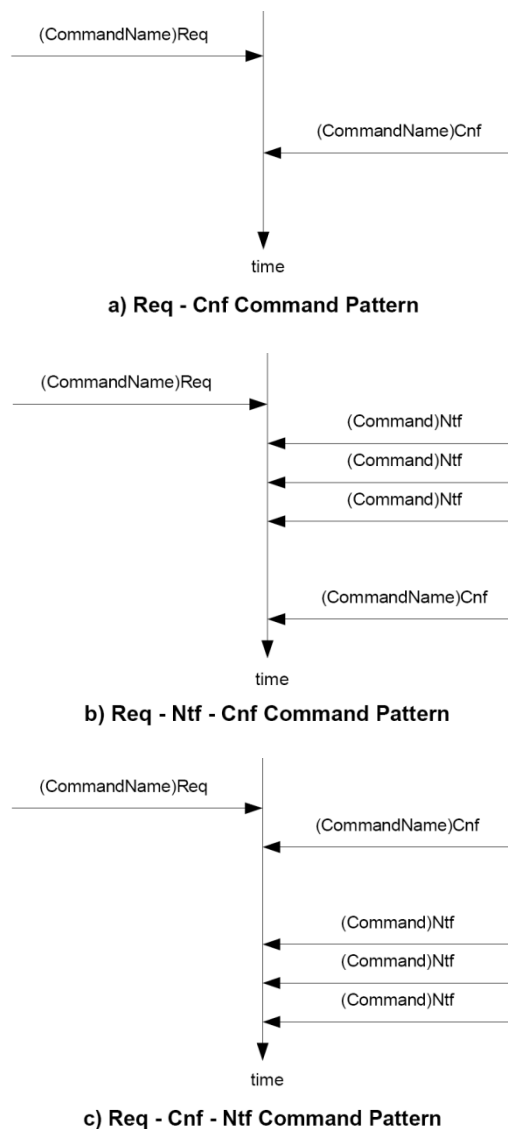


Figure 2 – Three different command patterns

5.2 Command types

5.2.1 General

The commands supported by the T-DMB common APIs can be categorized as follows.

- API-inquiry functions:
 - GetAPIVersion: Returns the API version.
 - Get T-DMBCapability: Returns API's T-DMB receiver capabilities and properties.
- Selecting an ensemble:
 - Tune: Tunes directly to a specified frequency.
 - Search: Searches for an ensemble.
- Accessing service directory:
 - SelectSI: Subscribes to service directory information.
 - GetEnsembleInfo: Gets information about a specified ensemble.
 - GetServiceInfo: Gets information about a specified service.
 - GetComponentInfo: Gets information about a specified component.
- Monitoring reception conditions:
 - SelectReceptionInfo: Subscribes to reception condition information.
- Selecting services:
 - SelectComponent: Starts or stops a service. In case of an audio/video service decoding of audio/video samples is started automatically. In case of a data service, the service can be accessed with the SelectObject command.
 - SelectApplication: Launches a T-DMB application.
 - SelectComponentStream: Gets access to the packet stream of the component.
- Selecting objects:
 - SelectObject: Requests data objects for delivery with or without automatic updating.
- Scanning for T-DMB services:
 - Scan: Scans a specified frequency range for T-DMB ensembles and updates the service directory.
- Miscellaneous:
 - OperationControl: accesses and modifies parameters of the receiver.
 - GetLocationInfo: retrieves location information from the receiver.

5.2.2 Get receiver capability

Figure 3 shows the get receiver capability. The T-DMB common APIs asks the T-DMB receiver for its capabilities.

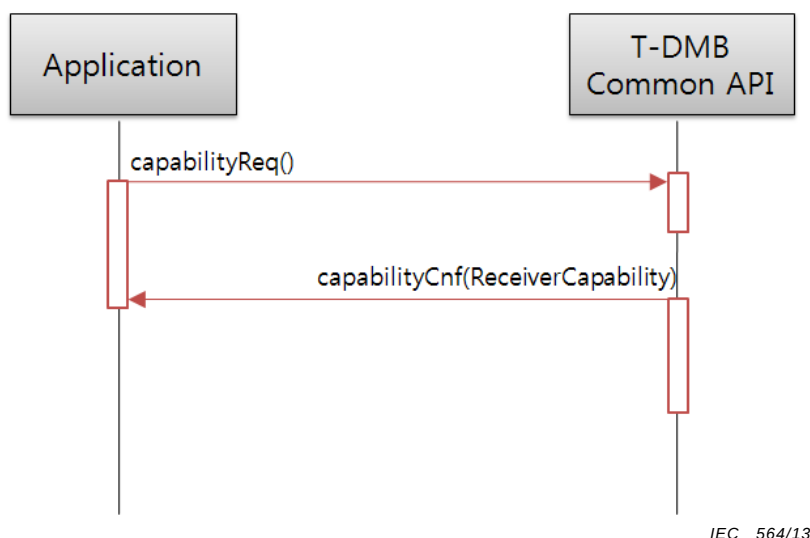


Figure 3 – Get receiver capability

void capabilityReq ()

The T-DMB Common APIs asks the T-DMB receiver for its capabilities.

Parameters

None

void capabilityCnf(ReceiverCapability)

T-DMB receiver provides its capabilities to the API.

5.2.3 Tuning

Figure 4 shows the tuning. The T-DMB receiver is tuned by calling tuneReq. The receiver tunes to the requested frequency and responds afterwards with tuneCnf confirmation. The confirmation contains information about the reception quality.

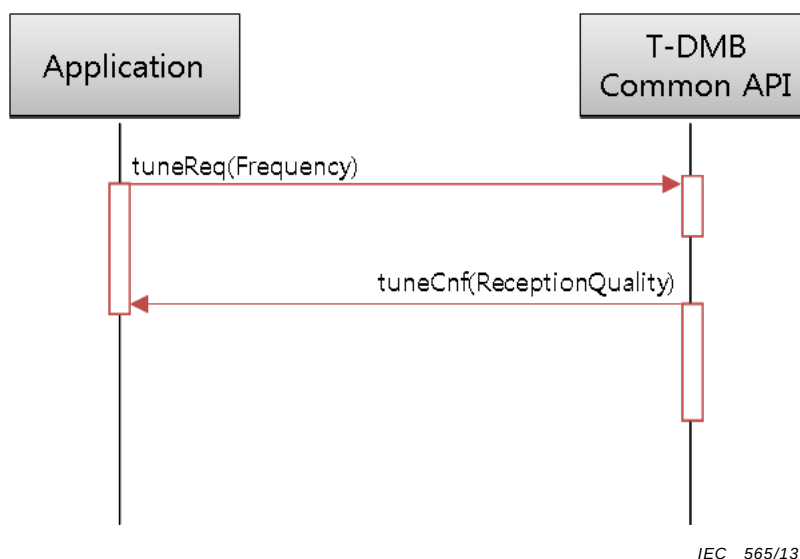


Figure 4 – Tuning

void tuneReq(int tuneFrequency,int transmissionMode)

The tuneReq request initiates the Tune command. The Tune command sets directly a specified T-DMB frequency. A T-DMB receiver shall be tuned to a T-DMB frequency and synchronized in order to get access to T-DMB services. A tuned T-DMB receiver tries automatically to synchronize on a T-DMB ensemble. The Tune command is used to select a

specified T-DMB frequency. Depending on the specification for the transmissionMode it is tested if a T-DMB ensemble can be detected. If the connected T-DMB receiver supports automatic detection the default setting for transmissionMode 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 T-DMB receiver.

Parameters

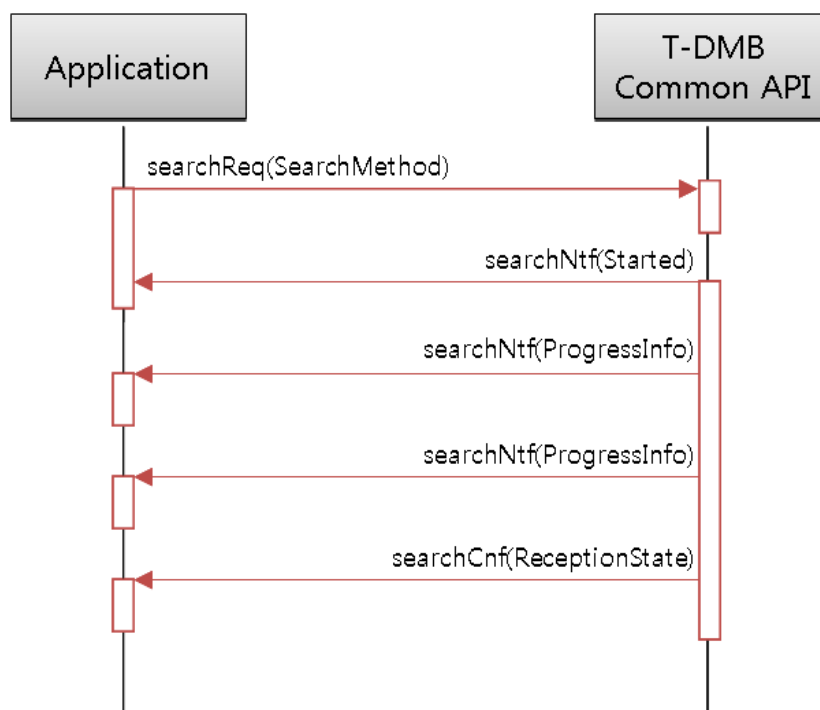
- tuneFrequency – This parameter specifies the frequency the T-DMB receiver will be tuned to in hertz.
- transmissionModes –
This parameter specifies the transmission modes a T-DMB receiver tests for T-DMB ensembles. The default value is T-DMBConstants.transmissionModeAutomatic which means that the receiver is automatically detecting the transmission mode. The parameter is a flag field supporting the following flags which can be specified together:
 - T-DMBConstants.transmissionModeAutomatic: The transmission mode is automatically detected. All other flags are ignored in this case.
 - T-DMBConstants.transmissionMode1: At the specified frequency it is tested if a T-DMB ensemble is sent in transmission mode 1.
 - T-DMBConstants.transmissionMode2: At the specified frequency it is tested if a T-DMB ensemble is sent in transmission mode 2.
 - T-DMBConstants.transmissionMode3: At the specified frequency it is tested if a T-DMB ensemble is sent in transmission mode 3.
 - T-DMBConstants.transmissionMode4: At the specified frequency it is tested if a T-DMB ensemble is sent in transmission mode 4.

void tuneCnf(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 T-DMB frequency and reception conditions. The Tune command is used to select a specified T-DMB frequency. The tuneReq request initiates the Tune command. The 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.

5.2.4 Searching

Figure 5 shows the searching. To search for some ensemble, the application calls searchReq. The T-DMB common APIs respond 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.



IEC 566/13

Figure 5 – Searching

```

void searchReq(
int searchMode,
int tables,
int startFrequency,
int stopFrequency,
int transmissionModes,
int notifications)
    
```

The searchReq request initiates a Search command. The Search command searches for a T-DMB ensemble according to a specified search mode. After a successful execution of the Search command a T-DMB ensemble has been found, the state Tuned is entered and the T-DMB receiver tries to synchronize automatically to the found T-DMB ensemble. The Search command is used to search for a T-DMB ensemble. The searchReq request initiates the search and specifies the frequencies and transmission modes to test. Additionally, the notifications that the T-DMB client gets can be specified 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 T-DMB ensemble at all is detected. In order to stop searching for a T-DMB ensemble the Tune command can be used which tunes the T-DMB 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 the Tune state enters the searching state (see Figure 5). If the previous state has been 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 T-DMB receiver is searching for a T-DMB ensemble. The default value is 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
 - T-DMBConstants.SearchModeAutomatic: default method

- T-DMBConstants.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.
- T-DMBConstants.SearchModeUp: The search direction is from low to high frequencies.
- T-DMBConstants.SearchModeDown: The search direction is from high to low frequencies.
- T-DMBConstants.SearchModeUseTables: The search is based on the specified frequency tables.
- T-DMBConstants.SearchModeUseFrequencyRange: The search is based on the specified frequency range.
- T-DMBConstants.SearchModeContinuous: The search is looping over the specified frequency range until a T-DMB 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 T-DMB ensembles. The parameter is a flag field supporting the following flags which can be specified together:
 - T-DMBConstants.searchCEPTFrequencyTableBandIII: The frequencies according to the CEPT frequency table for Band III are tested for T-DMB Ensembles.
 - T-DMBConstants.SearchCEPTFrequencyTableLBand: The frequencies according to the CEPT L-Band table are tested for T-DMB ensembles.
 - T-DMBConstants.SearchCanadaFrequencyTableLBand: The frequencies according to the Canadian L-Band table are tested for T-DMB ensembles.
- **transmissionModes** – This parameter specifies the transmission modes a T-DMB receiver tests for T-DMB ensembles. The default value is T-DMBConstants.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 T-DMB ensemble is sent in transmission mode 1.
 - transmissionMode2: At the specified frequency it is tested if a T-DMB ensemble is sent in transmission mode 2.
 - transmissionMode3: At the specified frequency it is tested if a T-DMB ensemble is sent in transmission mode 3.
 - transmissionMode4: At the specified frequency it is tested if a T-DMB 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 used only if 16 kHz step searching is specified as search mode.
 - notificationsTableEntry: With each table entry frequency a notification is sent. This is the default value.

void searchCnf(SearchCnfEvent e)

The searchCnf method finalizes a Search command and provides information about the command status, currently selected T-DMB frequency and current reception conditions. The Search command is used in order to search for a T-DMB ensemble according to a specified search mode. Searching for a T-DMB ensemble can take a large amount of time. The start of searching is indicated by a 'Started' searchNtf message. Other searchNtf messages inform a

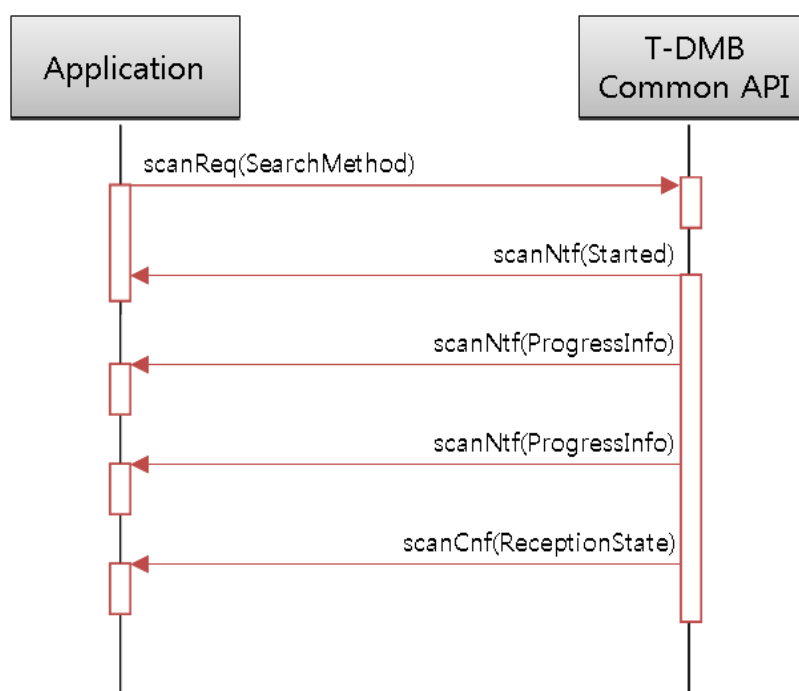
T-DMB client about search progress. It is finalized by delivery of the searchCnf message. It informs about the command status, the selected frequency and the synchronization state. No further searchNtf messages will be delivered after the searchCnf message.

void searchNtf(SearchNtfEvent e)

The SearchNtf event is sent after a search for a T-DMB 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 T-DMB 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.

5.2.5 Scanning

Figure 6 shows the scanning. The scanning means looking for ensembles in 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.



IEC 567/13

Figure 6 – Scanning

void scanReq(int searchMode, int tables, int startFrequency, int stopFrequency, int transmissionModes, int notifications)

The ScanReq request initiates a Scan command. The Scan command is used in order to perform a search for all available T-DMB 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= T-DMBConstants.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= T-DMBConstants.searchModeDown) the value of startFrequency is not allowed to be smaller than the value of stopFrequency.

Parameters

- **searchMode** – This parameter specifies the way the T-DMB receiver is searching for a T-DMB ensemble. The default value is T-DMBConstants.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:
 - T-DMBConstants.searchModeAutomatic: default method
 - T-DMBConstants.searchMode16kHzSteps: The frequency range is searched in 16 kHz steps.
 - T-DMBConstants.searchModeUp: The search direction is from low to high frequencies.
 - T-DMBConstants.searchModeDown: The search direction is from high to low frequencies.
 - T-DMBConstants.searchModeUseTables: The search is based on the specified frequency tables.
 - T-DMBConstants.searchModeUseFrequencyRange: The search is based on the specified frequency range.
 - T-DMBConstants.searchModeContinuous: The search is looping over the specified frequency range until a T-DMB 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 T-DMB ensembles. The parameter is a flag field supporting the following flags which can be specified together:
 - T-DMBConstants.searchCEPTFrequencyTableBandIII: The frequencies according to the CEPT frequency table for Band III are tested for T-DMB ensembles.
 - T-DMBConstants.searchCEPTFrequencyTableLBand: The frequencies according to the CEPT L-Band table are tested for T-DMB ensembles.
 - T-DMBConstants.searchCanadaFrequencyTableLBand: The frequencies according to the Canadian L-Band table are tested for T-DMB ensembles.
- **startFrequency** – This parameter specifies the start frequency at which the T-DMB receiver starts its search for T-DMB ensembles.
- **stopFrequency** – This parameter specifies the stop frequency at which the T-DMB receiver stops its search for T-DMB ensembles.
- **transmissionModes** – This parameter specifies the transmission modes a T-DMB receiver should look for T-DMB ensembles. The default value is T-DMBConstants.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:
 - T-DMBConstants.transmissionModeAutomatic: The transmission mode is automatically detected.
 - T-DMBConstants.transmissionMode1: At the specified frequency it is tested if a T-DMB ensemble is sent in transmission mode 1.
 - T-DMBConstants.transmissionMode2: At the specified frequency it is tested if a T-DMB ensemble is sent in transmission mode 2.

- T-DMBConstants.transmissionMode3: At the specified frequency it is tested if a T-DMB ensemble is sent in transmission mode 3.
- T-DMBConstants.transmissionMode4: At the specified frequency it is tested if a T-DMB 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:
 - T-DMBConstants.notificationsOff: No notifications are sent.
 - T-DMBConstants.notifications16kHzSteps: With each 16 kHz step a notification is sent.
 - T-DMBConstants.notificationsTableEntry: With each table entry frequency a notification is sent. This is the default value.

void scanCnf(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 T-DMB 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.

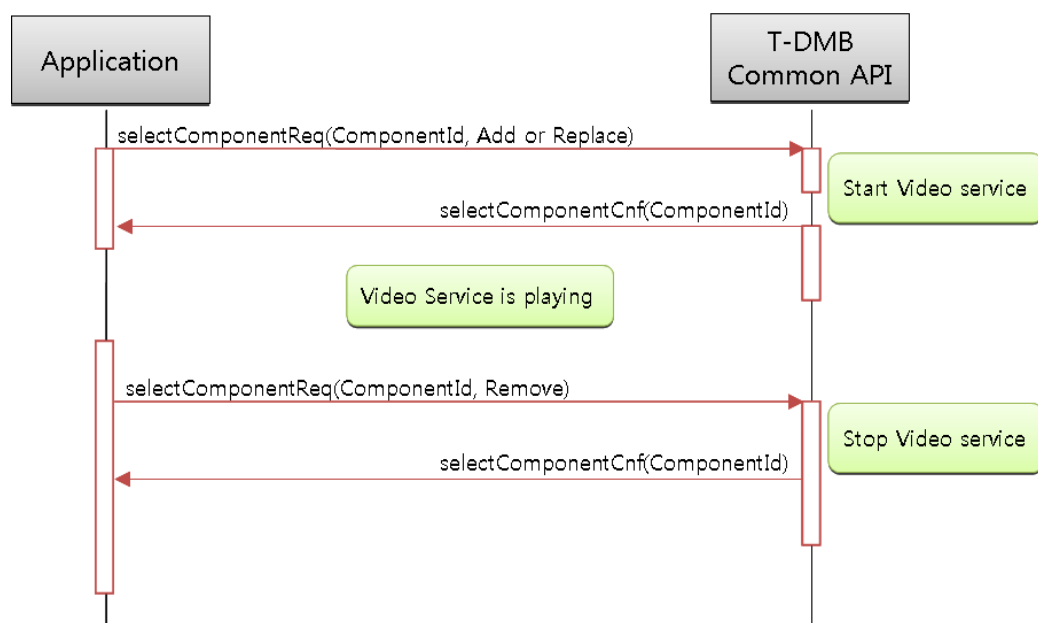
The ScanCnf message indicates that the Scan command is finished and informs about the current tune state. As a result of performing the Scan command the service information database is filled with information. If a SI subscription is running several SINtf messages are delivered to the connected application.

void scanNtf(ScanNtfEvent e)

The ScanNtf message is sent after a search for all available T-DMB 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 T-DMB 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.

5.2.6 Selecting a T-DMB service

Figure 7 shows the selecting a T-DMB service. An audio/video service is started with the SelectComponent command. The application calls selectComponentReq passing the identifier of the audio/video component. The T-DMB common APIs will start the audio/video service and send back a confirmation. To stop this audio/video service, the application calls selectComponentReq again now specifying that the component has to be removed. When the T-DMB common APIs respond with a confirmation, the audio/video service has been stopped.



IEC 568/13

Figure 7 – Selecting a T-DMB service

void selectComponentReq(ComponentId id, int selectionMode)

The `selectComponentReq` request initiates the `SelectComponent` command. The `SelectComponent` command starts or stops an application delivered in a T-DMB component. The `SelectComponent` command allows to start or stop applications delivered in T-DMB components. In general, more than one component of the same T-DMB 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 T-DMB 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 well. 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 T-DMB component which is to be selected. If all component selections should be removed (set `selectionMode` to `T-DMBConstants.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:
 - `T-DMBConstants.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.

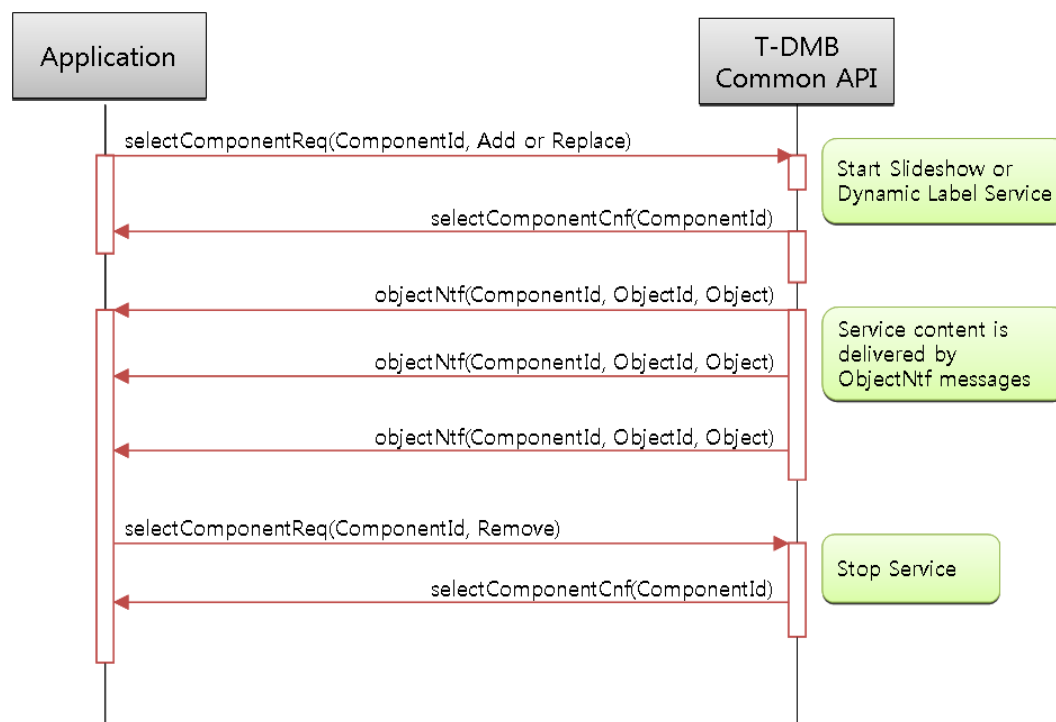
- T-DMBConstants.selectionModeAdd: The application delivered by the specified component is to be started. Other selected components are not affected.
- T-DMBConstants.selectionModeRemove: The selection of the specified component is stopped.
- T-DMBConstants.selectionModeRemoveAll: All existing component selections are removed. Set servceld to null in this case.

void selectComponentCnf(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 to stop applications delivered in T-DMB components. In general, more than one component of the same T-DMB 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 T-DMB 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.

5.2.7 Selecting a slideshow or a dynamic label service

Figure 8 shows the selection of a slideshow or a dynamic label service. An application selects a slideshow or a dynamic label service with the SelectComponent command. When the request selectComponentReq with the respective service identifier is issued, the service gets 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 by setting selectionMode to selectionModeRemove. The removal of the service will be confirmed.



IEC 569/13

Figure 8 – Selecting a slideshow or a dynamic label service

void objectNtf(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 T-DMB client. objectNtf is used to deliver a selected object to the connected T-DMB client. Depending on the request mode the object is delivered only once or more than once in case of updates. If the object cannot be delivered in-time as indicated by a call to selectObjectCnf, then objectNtf informs about the delay. If transmission of a selected object is stopped, objectNtf informs about the termination of the object transmission and the object selection. It is possible that a T-DMB component is removed from a T-DMB 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.

5.2.8 Selecting a broadcast website service

Figure 9 shows the selection of a broadcast website service. To run a broadcast website service the component has to be selected. This is accomplished by calling selectComponentReq with the respective service identifier. The start of the service is confirmed by the T-DMB common APIs. The actual objects of the service are retrieved with the selectObject command. Usually, the start object is demanded first. For that, a selectObjectReq request is issued with the service identifier of the component and the object identifier of the start object. The T-DMB common APIs will send back a confirmation including the likely access time. The actual object is received with an objectNtf notification. All other objects of the service are requested and delivered similarly. The service is stopped 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).

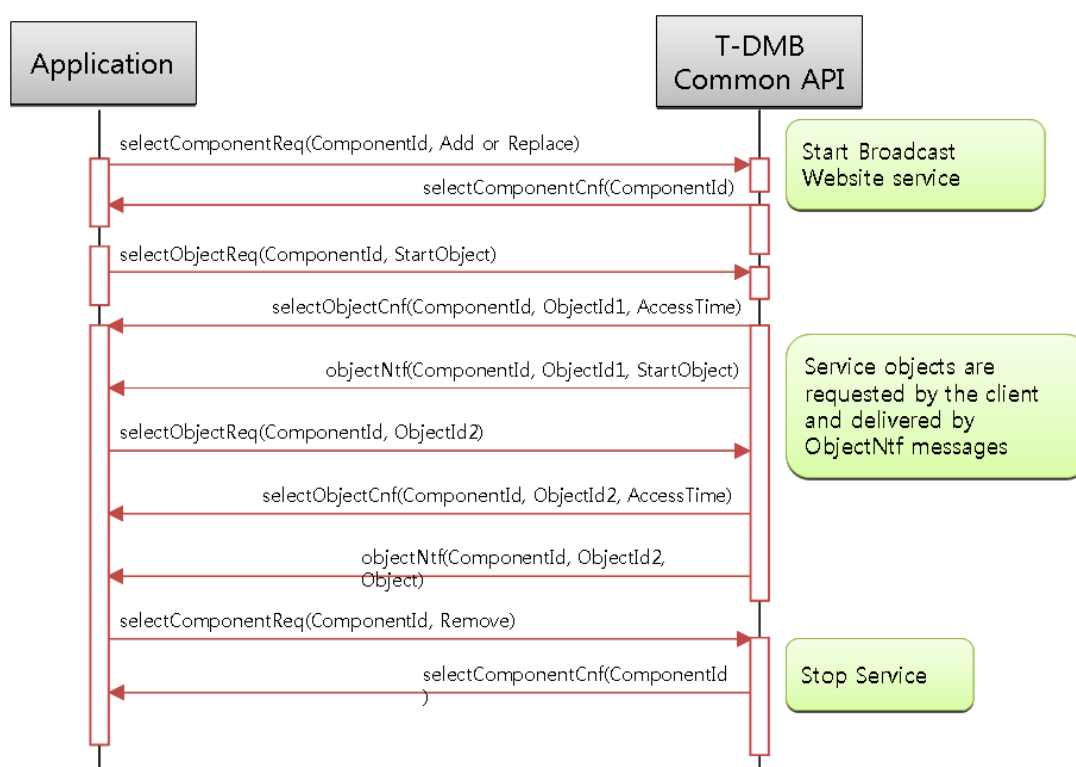


Figure 9 – Selecting a broadcast website service

IEC 570/13

```

void selectObjectReq(
ComponentId id,
ObjectId objectId,
int requestMode,
boolean replaceSelections,
int deliveryMode,

```

int cacheHint)

The selectObjectReq request initiates the SelectObject command. The SelectObject command selects an object from a selected T-DMB component. This includes requesting an object from a data component, delivery after reception and notification of updates as long as the object is selected. 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 to remove a component from a T-DMB 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 to remove an object from a 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:
 - T-DMBConstants.requestModeOff: This is used in order to stop the selection of objects which are requested with the request mode T-DMBConstants.requestModeUpdate. It is not needed for objects which are requested with the T-DMBConstants.requestModeOnce flag except when a SelectObjectReq is pending and the delivery is no longer wanted.
 - T-DMBConstants.requestModeOnce: The object is requested for one-time delivery. After the first reception from the broadcast channel the object is delivered to the connected T-DMB client. The client is not notified about new versions.
 - T-DMBConstants.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 whether 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:
 - T-DMBConstants.deliveryModeComplete: Only the complete object is delivered to the T-DMB client.
 - T-DMBConstants.deliveryModePartial: The object may be delivered in parts.
- **cacheHint** – This parameter specifies a hint for caching of the selected object.

void selectObjectCnf(SelectObjectCnfEvent e)

The SelectObjectCnf method finalizes the SelectObject command. The SelectObject command selects an object from a selected T-DMB component. This includes requesting an object from a data component, delivery after reception and notification of updates as long as the object is selected. 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 can be selected simultaneously as well as from more than one component. 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 to remove a component from a T-DMB ensemble. 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.

5.2.9 Get T-DMB service information

Figure 10 shows the Get T-DMB service information. The application can also use the ServiceInfo command to retrieve the respective T-DMB service information objects. It has to specify the service identifier in the siReq request. The confirmation will then contain the requested object.

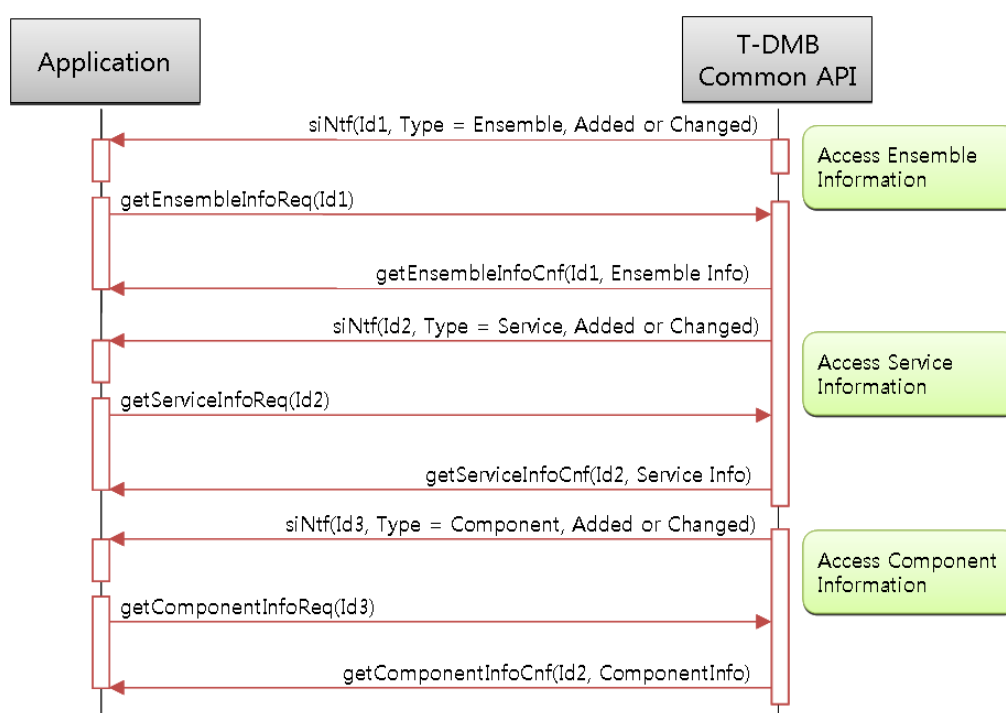


Figure 10 – Get T-DMB service information

IEC 571/13

void getEnsembleInfoReq(EnsembleId id)

The getEnsembleInfoReq method initiates a GetEnsembleInfo command. The GetEnsembleInfo command requests information about the specified T-DMB ensemble. The GetEnsembleInfo command provides a T-DMB client with information about a specified T-DMB 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 T-DMB ensemble.

void getServiceInfoReq(ServiceId id)

The getServiceInfoReq requests initiates a GetServiceInfo command. The GetServiceInfo command requests information about a specified T-DMB service. The GetServiceInfo command provides a T-DMB client with information about a specified T-DMB 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 T-DMB service.

void GetComponentInfoReq(ComponentId id)

The GetComponentInfoReq request initiates a GetComponentInfo command. The GetComponentInfo command requests information about a specified T-DMB component. The GetComponentInfo command provides a T-DMB client with information about a specified T-DMB component, e.g. label, language, and so on. The command is initiated calling GetComponentInfoReq and is finished by a call to GetComponentInfoCnf.

Parameters

- **id** – This parameter is a handle identifying the T-DMB component.

void getEnsembleInfoCnf(GetEnsembleInfoCnfEvent e)

The GetEnsembleInfoCnf method finalizes the GetEnsembleInfo command and delivers information about a requested T-DMB ensemble to a T-DMB client. The GetEnsembleInfo command provides a T-DMB client with information about a specified T-DMB 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.

void getServiceInfoCnf(GetServiceInfoCnfEvent e)

A call to the getServiceInfoCnf method finalizes the GetServiceInfo command and delivers information about a requested T-DMB Service to a T-DMB client. The GetServiceInfo command provides a T-DMB client with information about a specified T-DMB 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.

void GetComponentInfoCnf(GetComponentInfoCnfEvent e)

The GetComponentInfoCnf message finalizes the GetComponentInfo command and delivers information about a requested T-DMB component to a T-DMB client. The GetComponentInfo command provides a T-DMB client with information about a specified T-DMB 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.

5.2.10 Monitoring reception qualities

Figure 11 shows the monitoring reception qualities. 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 receives receptionInfoNtf notifications as long as the monitoring is not stopped.

(selectionReceptionInfoReq (Off)).

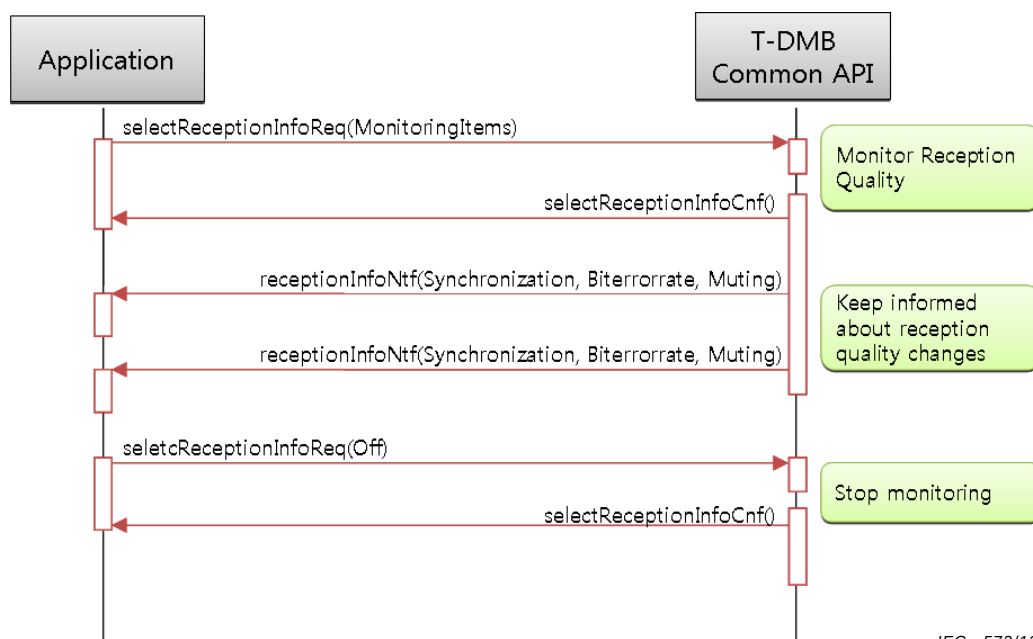


Figure 11 – Monitoring reception qualities

IEC 572/13

**void selectReceptionInfoReq(
boolean synchronizationNotification,
boolean bitErrorRateNotifications,
boolean muteStateNotifications,
boolean requestOnce)**

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, bit-error-rate and audio decoder muting. The SelectReceptionInfo command allows a T-DMB client to subscribe for state change notifications concerning reception conditions in terms of synchronization, bit-error-rate 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 T-DMB client receives ReceptionInfoNtf notifications when state changes occur.

Parameters

- **synchronizationNotification** – This parameter specifies if the calling client is notified about state changes concerning T-DMB signal synchronization. If the parameter is set to true (default) notifications are provided, if it is set to false no notifications are provided.
- **bitErrorRateNotifications** – This parameter specifies if the calling client is notified about state changes concerning the bit-error-rate. 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 T-DMB client informed by one and only one receptionInfoNtf call.

void selectReceptionInfoCnf(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 T-DMB client to subscribe for state change notifications concerning reception conditions in terms of synchronization, bit-error-rate 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 T-DMB client receives receptionInfoNtf calls when state changes occur.

void receptionInfoNtf(ReceptionInfoNtfEvent e)

The receptionInfoNtf method is called as a consequence of subscribing to state changes in synchronization, bit-error-rate and audio decoder muting. receptionInfoNtf indicates that the synchronization state, bit-error-rate or mute state has changed (see 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).

Annex A (informative)

Examples of the classes used in T-DMB APIs

In this annex, examples of the classes used in T-DMB APIs are described.

```
SearchCnfEvent(  
T-DMBSource source,  
int result,  
int tuneState,  
int tuneFrequency,  
int transmissionMode,  
int synchronizationState)
```

```
SearchNtfEvent(  
T-DMBSource source,  
int tuneFrequency,  
int notifications)
```

```
ScanNtfEvent(  
T-DMBSource source,  
int tuneFrequency,  
int notifications)
```

```
ScanCnfEvent(  
T-DMBSource source,  
int result,  
int tuneState,  
int tuneFrequency,  
int transmissionModes,  
int noOfEnsemblesFound)
```

```
GetEnsembleInfoCnfEvent(  
T-DMBSource source,  
int result,  
EnsembleInfo ensembleInfo)
```

```
GetServiceInfoCnfEvent(  
T-DMBSource source,  
int result,  
ServiceInfo serviceInfo)  
GetComponentInfoCnfEvent(  
T-DMBSource source,  
int result,  
ComponentInfo componentInfo)
```

```
SelectReceptionInfoCnfEvent(  
T-DMBSource source,  
int result,  
boolean synchronizationNotifications,  
boolean bitErrorRateNotifications,  
boolean muteStateNotifications)
```

```
ReceptionInfoNtfEvent(  
T-DMBSource source,  
int updateFlags,  
int synchronizationState,  
int bitErrorRateState,
```

int muteState)

SelectComponentCnfEvent(
T-DMBSource source,
int result,
ComponentId componentId,
int selectionMode)

ObjectNtfEvent(
T-DMBSource source,
ComponentId componentId,
ObjectId objectId,
int selectionState,
T-DMBObject object)

SelectObjectCnfEvent(
T-DMBSource source,
int result,
ComponentId componentId,
ObjectId objectId,
int requestMode,
boolean replaceSelections,
Date accessTime)

ComponentInfo(
ComponentId id,
int type,
byte[] data,
boolean isPrimary,
ServiceId[] parentIds,
int accessControlSystem,
boolean hasLabel,
Label label,
boolean hasLanguage,
int language,
boolean hasStartObjectId,
ObjectId startObjectId,
boolean hasObjectDirectoryId,
ObjectId objectDirectoryId,
boolean hasAudioComponent,
ComponentId audioComponent,
boolean hasBitrate,
int bitrate)

ServiceInfo(
ServiceId id,
int type,
EnsembleId parent,
ComponentId[] componentIds,
boolean isLocal,
int accessControlSystem,
boolean hasLabel,
Label label,
boolean hasLanguage,
int language,
boolean hasIsPrimary,
boolean isPrimary,
boolean hasRegionId,
int regionId,
boolean hasRegionLabel,
Label regionLabel,
boolean hasStaticProgrammeType,

```
ProgrammeType staticProgrammeType,  
boolean hasDynamicProgrammeType,  
ProgrammeType dynamicProgrammeType,  
boolean hasProgrammeNumber,  
ProgrammeNumber programmeNumber,  
boolean hasTimeOffset,  
int timeOffset,  
boolean hasAnnouncementSupport,  
AnnouncementSupport announcementSupport,  
boolean hasCountry,  
int country)
```

```
EnsembleInfo(  
EnsembleId id,  
ServiceId[] serviceIds,  
int frequency,  
int transmissionMode,  
boolean hasDate,  
Date date,  
boolean hasLabel,  
Label label,  
boolean hasCountry,  
int country)
```

Bibliography

IEC 62104:2003, *Characteristics of DAB receivers*

ISO 10486:1992, *Passenger cars – Car radio identification number (CRIN)*

ETSI EN 50094:1992, *Access control system for the MAC/packet family: Eurocrypt*
ETSI TS 101 993 v1.1.1 *Digital Audio Broadcasting (DAB); A Virtual Machine for DAB: DAB Java Specification*

SOMMAIRE

AVANT-PROPOS	31
1 Domaine d'application	33
2 Références normatives	33
3 Abréviations	33
4 Vue d'ensemble de l'API commune de la T-DMB	34
4.1 Vue d'ensemble d'un récepteur T-DMB.....	34
4.2 Bloc ASIC d'un récepteur T-DMB.....	35
4.3 Bloc processeur hôte.....	35
4.3.1 Généralités.....	35
4.3.2 Sous-bloc pilote T-DMB (couche d'abstraction matérielle)	35
4.3.3 Sous-bloc logiciel spécifique ASIC T-DMB.....	36
4.3.4 Sous-bloc API communes T-DMB	36
4.3.5 Sous-bloc intergiciel du récepteur T-DMB.....	36
4.4 Bloc interface matérielle	36
5 Description de l'API	36
5.1 API communes T-DMB	36
5.2 Types de commande	37
5.2.1 Généralités.....	37
5.2.2 Obtention des capacités du récepteur.....	38
5.2.3 Syntonisation.....	39
5.2.4 Recherche.....	41
5.2.5 Balayage	43
5.2.6 Sélection d'un service T-DMB	46
5.2.7 Sélection d'un diaporama ou d'un service d'étiquettes dynamiques.....	48
5.2.8 Sélection d'un service de site Web à large diffusion	49
5.2.9 Informations de service Get T-DMB	52
5.2.10 Surveillance de la qualité de réception	53
Annexe A (informative) Exemples de classes utilisées dans les API T-DMB.....	56
Bibliographie.....	59
Figure 1 – Schéma par blocs d'un récepteur T-DMB type.....	35
Figure 2 – Trois configurations de commande différentes	37
Figure 3 – Obtention des capacités du récepteur	39
Figure 4 – Syntonisation	40
Figure 5 – Recherche	41
Figure 6 – Balayage.....	44
Figure 7 – Sélection d'un service T-DMB	47
Figure 8 – Sélection d'un diaporama ou d'un service d'étiquettes dynamiques	49
Figure 9 – Sélection d'un service de site Web à large diffusion	50
Figure 10 – Informations de service Get T-DMB.....	52
Figure 11 – Surveillance de la qualité de réception	54

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

RÉCEPTEURS POUR DIFFUSION MULTIMÉDIA NUMÉRIQUE TERRESTRE (T-DMB) –

Partie 3: API commune

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de brevet. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale CEI 62516-3 a été établie par le domaine technique 1: Terminaux pour les contenus audio, vidéo et services de données, du Comité d'études 100 de la CEI: Systèmes et équipements audio, vidéo et services de données.

Le texte de cette norme est issu des documents suivants:

CDV	Rapport de vote
100/2020/CDV	100/2110/RVC

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 62516, présentées sous le titre général *Récepteurs pour diffusion multimédia numérique terrestre (T-DMB)*, peut être consultée sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "*colour inside*" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

RÉCEPTEURS POUR DIFFUSION MULTIMÉDIA NUMÉRIQUE TERRESTRE (T-DMB) –

Partie 3: API commune

1 Domaine d'application

La présente partie de la CEI 62516 décrit l'interface de programmation d'application (API) commune T-DMB. Elle fournit une plate-forme logicielle qui, lorsqu'elle est combinée avec le système d'exploitation O/S T-DMB, constitue une interface universelle pour les programmes d'application. Cette interface permet d'écrire des programmes d'application de telle manière qu'ils puissent être exécutés sur n'importe quelle unité de récepteur T-DMB, comme décrit dans les CEI 62516-1:2009 et CEI 62516-2:2011 quel que soit son fabricant.

La présente partie de la CEI 62516 définit également un environnement logiciel permettant à plusieurs programmes d'application d'être interopérables sur une unité de récepteur unique, en partageant les ressources fixes du récepteur. Elle fournit également un ensemble d'interfaces utilisées par l'intergiciel T-DMB et le logiciel spécifique à un ASIC (Circuit intégré à application spécifique).

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

CEI 62516-1:2009, *Récepteurs pour diffusion multimédia numérique terrestre (T-DMB) – Partie 1: Exigences fondamentales*

CEI 62516-2:2011, *Récepteurs pour diffusion multimédia numérique terrestre (T-DMB) – Partie 2: Services de données interactifs utilisant le BIFS*

ETSI EN 300 401 V1.3.3, *Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers*

3 Abréviations

CAN	Convertisseur analogique-numérique
API	Interface de programmation d'application (en anglais: Application Programming Interface)
ASIC	Circuit intégré spécifique à une application (en anglais: Application Specific Integrated Circuit)
FIC	Canal d'information rapide (en anglais: Fast Information Channel)
HAL	Couche d'abstraction matérielle (en anglais: Hardware Abstraction Layer)
ISR	Routine de service d'interruption (en anglais: Interrupt Service Routine)
MAC	Contrôle d'accès au support (en anglais: Media Access Control)
PAD	Données associées au programme (en anglais: Program Associated Data)
RF	Radiofréquence

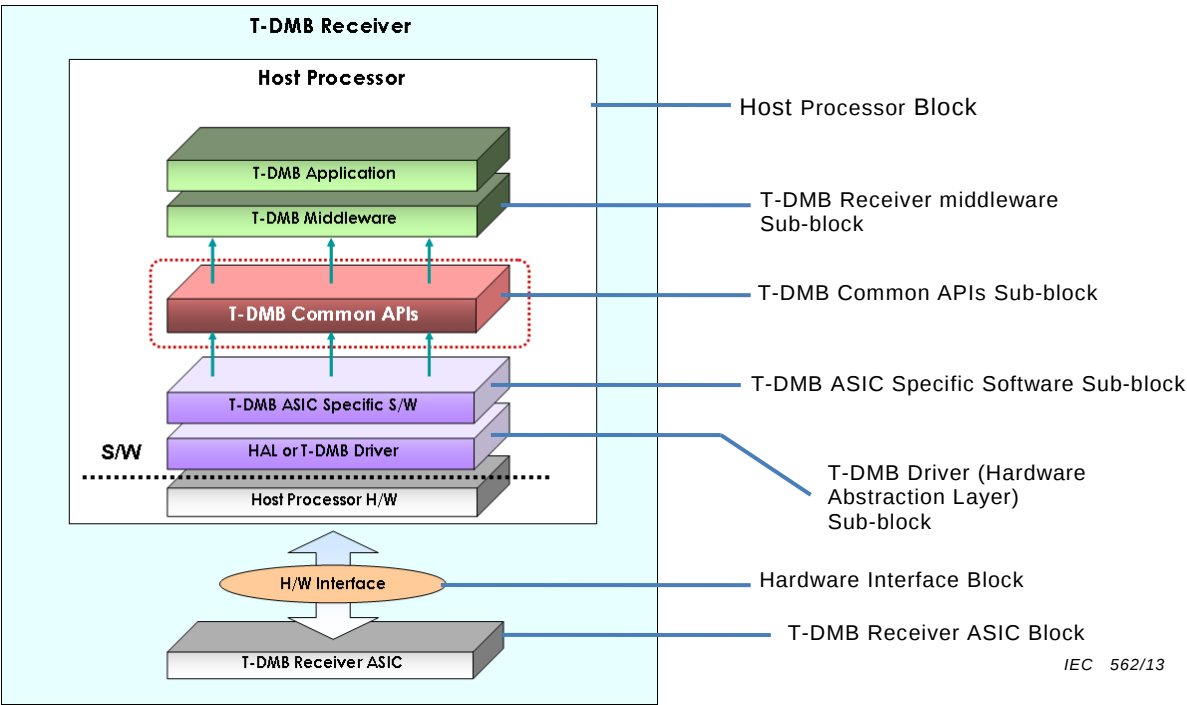
- R-S Reed Solomon
- SDIO Entrée/sortie numérique sécurisée (en anglais: Secure Digital Input/Output)
- T-DMB Diffusion multimédia numérique terrestre (en anglais: Terrestrial-Digital Multimedia Broadcasting)

4 Vue d'ensemble de l'API commune de la T-DMB

4.1 Vue d'ensemble d'un récepteur T-DMB

Un récepteur T-DMB procure la fonctionnalité de dispositif spécifiée dans les récepteurs pour T-DMB (voir la CEI 62516-1:2009 et la CEI 62516-2:2011). La Figure 1 est un schéma par blocs d'un récepteur T-DMB type. Pour le récepteur T-DMB représenté à la Figure 1, seuls les blocs conformes au domaine d'application de cette norme sont représentés.

La Figure 1 représente également l'API commune de la T-DMB par rapport au schéma par blocs du récepteur T-DMB.



Légende

Anglais	Français
T-DMB Receiver	Récepteur T-DMB
Host Processor	Processeur hôte
T-DMB Application	Application T-DMB
T-DMB Middleware	Intergiciel T-DMB
T-DMB Common APIs	API communes T-DMB
T-DMB ASIC Specific APIs	API spécifiques ASIC T-DMB
HAL or T-DMB Driver	Pilote HAL ou T-DMB
Host Processor H/W	H/W processeur hôte
H/W Interface	Interface H/W
T-DMB Receiver ASIC	ASIC récepteur T-DMB
Host Processor Block	Bloc processeur hôte

Anglais	Français
T-DMB Receiver middleware Sub-block	Sous-bloc Intergiciel récepteur T-DMB
T-DMB Common APIs Sub-block	Sous-bloc API communes T-DMB
T-DMB ASIC Specific Software Sub-block	Sous-bloc logiciel spécifique ASIC T-DMB
T-DMB Driver (Hardware Abstraction Layer) Sub-block	Sous-bloc pilote T-DMB (couche d'abstraction matérielle)
Hardware Interface Block	Bloc interface matérielle
T-DMB Receiver ASIC Block	Bloc ASIC récepteur T-DMB

Figure 1 – Schéma par blocs d'un récepteur T-DMB type

4.2 Bloc ASIC d'un récepteur T-DMB

Le bloc ASIC d'un récepteur T-DMB représente les circuits à semiconducteur procurant la fonctionnalité de démodulation d'un signal T-DMB et de récupération des données transportées par la couche physique de la T-DMB. Ce bloc procure des fonctionnalités telles que l'étage d'entrée RF, le CAN, l'estimation du temps et de la fréquence, l'estimation du canal, le décodage de Viterbi, etc. En somme, ce bloc assure la mise en œuvre complète de la couche physique selon l'ETSI EN 300 401 v1.3.3. Selon la mise en œuvre, ce bloc peut également procurer une fonctionnalité de décodage externe (par exemple, décodage R-S et/ou désentrelacement convolutionnel).

4.3 Bloc processeur hôte

4.3.1 Généralités

Le bloc processeur hôte représente la fonctionnalité de la T-DMB procurée par le processeur hôte dans un dispositif T-DMB. En d'autres termes, celui-ci représente les circuits du processeur hôte et la mise en œuvre logicielle résidant dans le processeur hôte. Le bloc processeur hôte récupère et traite les informations de T-DMB obtenues depuis le bloc ASIC du récepteur T-DMB. Les informations de T-DMB récupérées sont constituées d'informations de configuration de multiplex reçues sur le FIC (Canal d'information rapide), contenu qui est reçu sur le canal de service principal. Ce bloc communique avec le bloc ASIC du récepteur T-DMB pour récupérer les informations reçues du signal T-DMB. Le bloc processeur hôte est constitué des sous-blocs fonctionnels suivants.

4.3.2 Sous-bloc pilote T-DMB (couche d'abstraction matérielle)

Le bloc pilote T-DMB, ou couche d'abstraction matérielle (HAL) représente le logiciel au niveau pilote dans le processeur principal directement interfacé avec le bloc ASIC du récepteur T-DMB. Le sous-bloc pilote T-DMB assure les fonctions de contrôleur (par exemple, activation ou désactivation du bloc ASIC du récepteur T-DMB) et les fonctions d'échange de données (par exemple, récupération des données depuis le bloc ASIC du récepteur T-DMB ou acheminement des caractéristiques d'un sous-canal à recevoir) pour des circuits ASIC de récepteur T-DMB donnés. Le logiciel pilote T-DMB est spécifique au type de mécanisme d'interface matérielle existant entre le bloc processeur hôte et le bloc ASIC du récepteur T-DMB.

Par exemple, le logiciel pilote T-DMB sera différent selon que l'interface matérielle entre le processeur principal et le bloc ASIC du récepteur T-DMB est pilotée par interruption, mise en œuvre avec adresse/registres mappés en mémoire ou interface de transaction par paquets telle que SDIO. On peut citer comme exemples de tâches exécutées par le sous-bloc pilote T-DMB:

- Interactions matérielles, telles qu'une initialisation, des déclencheurs de mise en sommeil ou de réveil.
- Échange de données avec des circuits tel que le vidage de circuits tampons dans la mémoire principale ou la fourniture d'une mise en œuvre ISR.

Les fonctions du logiciel pilote T-DMB sont étroitement liées aux circuits ASIC du récepteur T-DMB et sont considérées comme d'une nature sensible au temps. Le logiciel pilote T-DMB reçoit donc généralement une priorité supérieure par rapport aux autres sous-blocs présentés. Par exemple, le pilote T-DMB exécute les tâches de récupération des données reçues par le bloc ASIC du récepteur T-DMB ou de commande au bloc ASIC du récepteur T-DMB de se synchroniser sur une fréquence comme demandé par la couche application.

4.3.3 Sous-bloc logiciel spécifique ASIC T-DMB

Le sous-bloc logiciel spécifique ASIC T-DMB fournit la fonctionnalité de la couche MAC qui n'est pas couverte par le sous-bloc pilote T-DMB. En fonction de la division de la fonctionnalité de la couche MAC parmi différents sous-blocs, il peut assurer une fonctionnalité complète ou partielle de la couche MAC. On s'attend au moins à ce que le sous-bloc logiciel spécifique ASIC T-DMB fournisse une fonctionnalité de couche MAC de haut niveau qu'il n'est pas possible de le déléguer au sous-bloc pilote T-DMB. Il s'interface avec l'intergiciel du récepteur T-DMB en utilisant les API communes T-DMB.

4.3.4 Sous-bloc API communes T-DMB

Le sous-bloc API communes T-DMB définit les interfaces permettant au sous-bloc logiciel spécifique ASIC T-DMB de communiquer avec l'intergiciel du récepteur T-DMB. Tout intergiciel de récepteur T-DMB adhérant aux interfaces définies par les API communes T-DMB fonctionnera avec n'importe quel sous-bloc logiciel spécifique ASIC T-DMB adhérant également à ces interfaces. De plus amples détails concernant cette interface sont fournis dans le reste de la présente norme.

4.3.5 Sous-bloc intergiciel du récepteur T-DMB

Le sous-bloc intergiciel du récepteur T-DMB communique avec le sous-bloc logiciel spécifique ASIC T-DMB en utilisant les API communes T-DMB. L'intergiciel du récepteur T-DMB met en œuvre la couche contrôle et flux et assure l'interface avec les protocoles de la couche application. Il déclenche le logiciel spécifique ASIC T-DMB pour recevoir le contenu spécifié demandé par la couche application. Il agit sur les notifications ou le contenu fournis par le logiciel spécifique ASIC T-DMB. Il délivre tout contenu reçu depuis le logiciel spécifique ASIC T-DMB aux protocoles de la couche application.

4.4 Bloc interface matérielle

Le bloc interface matérielle représente le mécanisme d'interface matérielle qui existe entre le bloc processeur hôte et le bloc ASIC du récepteur T-DMB. Cette interface assure la fonctionnalité de communication et d'échange de données. Le sous-bloc pilote T-DMB utilise ce bloc pour échanger des commandes et des données avec le bloc ASIC du récepteur T-DMB. Le bloc interface matérielle peut être constitué d'une quelconque interface souhaitée, par exemple une interface de bus propriétaire ou une interface normalisée (par exemple SDIO).

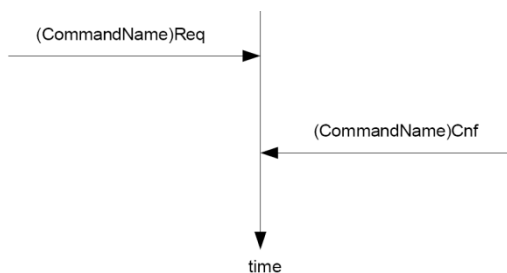
5 Description de l'API

5.1 API communes T-DMB

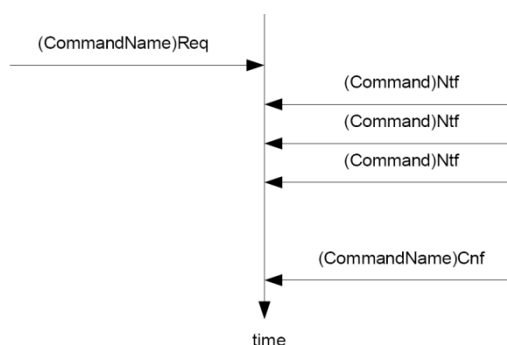
Cet article fournit une description détaillée de chaque API commune de T-DMB. Les détails du prototype de fonction API sont fournis avec des types définis dont ont besoin les API communes T-DMB.

Les commandes sont exécutées en envoyant des demandes, des confirmations et des notifications. La Figure 2 représente trois configurations de commande différentes. Celles-ci sont utilisées dans les API communes T-DMB.

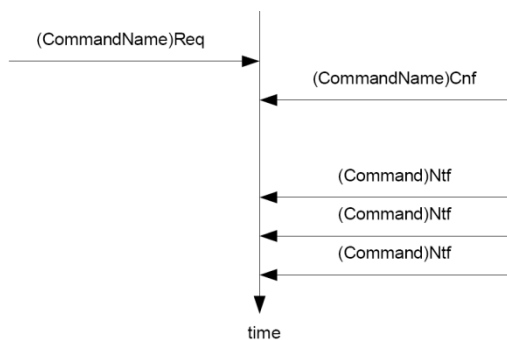
Si des commandes sont entrelacées, ce qui signifie deux commandes s'exécutant en même temps, une séquence arbitraire de types de message est possible.



a) Req - Cnf Command Pattern



b) Req - Ntf - Cnf Command Pattern



c) Req - Cnf - Ntf Command Pattern

IEC 563/13

Légende

Anglais	Français
a) Req - Cnf Command Pattern	a) Configuration de commande Req - Cnf
time	temps
b) Req - Ntf - Cnf Command Pattern	b) Configuration de commande Req - Ntf - Cnf
time	temps

Figure 2 – Trois configurations de commande différentes

5.2 Types de commande

5.2.1 Généralités

Les commandes prises en charge par les API communes T-DMB peuvent être classées comme suit.

- Fonctions API-inquiry:
 - GetAPIVersion: Retourne la version de l'API.

- Get T-DMBCapability: Retourne les capacités et les propriétés du récepteur T-DMB des API.
- Sélection d'un Ensemble:
 - Tune: Syntonisation directe sur une fréquence spécifiée.
 - Search: Recherche d'un Ensemble.
- Accès au répertoire de services:
 - SelectSI: Abonnement aux informations du répertoire de services.
 - GetEnsembleInfo: Obtention d'informations relatives à un ensemble spécifié.
 - GetServiceInfo: Obtention d'informations relatives à un service spécifié.
 - GetComponentInfo: Obtention d'informations relatives à un composant spécifié.
- Surveillance des conditions de réception:
 - SelectReceptionInfo: Abonnement aux informations de conditions de réception.
- Sélection de services:
 - SelectComponent: Démarrage ou arrêt d'un service. Dans le cas d'un service audio/vidéo, le décodage des échantillons audio/vidéo démarre automatiquement. Dans le cas d'un service de données, on peut accéder au service à l'aide de la commande SelectObject.
 - SelectApplication: Lancement d'une application T-DMB.
 - SelectComponentStream: Obtention de l'accès aux flux de paquets du composant.
- Sélection d'objets:
 - SelectObject: Demande d'objets de données pour fourniture avec ou sans mise à jour automatique.
- Balayage de services T-DMB
 - Scan: Balayage d'une plage de fréquences spécifiée pour rechercher des ensembles T-DMB et mise à jour du répertoire de services.
- Divers:
 - OperationControl: accès et modification des paramètres du récepteur.
 - GetLocationInfo: récupération des informations de localisation depuis le récepteur.

5.2.2 Obtention des capacités du récepteur

La Figure 3 représente l'obtention des capacités du récepteur. L'API commune de T-DMB demande au récepteur T-DMB ses capacités.

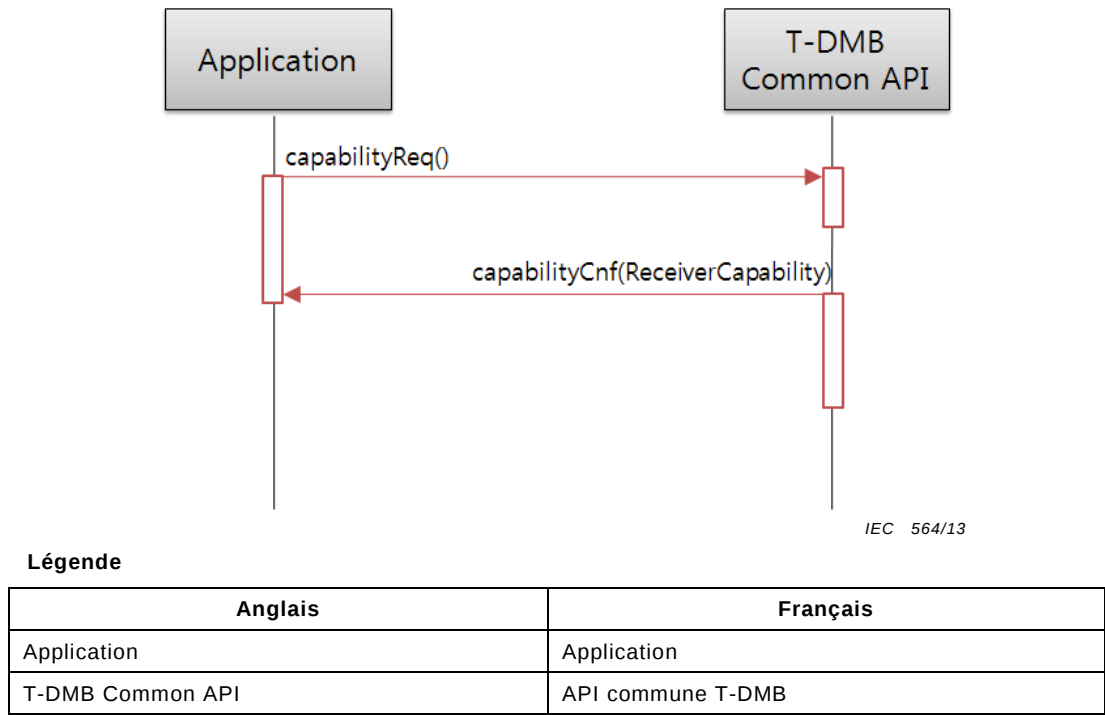


Figure 3 – Obtention des capacités du récepteur

void capabilityReq ()

L'API commune de T-DMB demande au récepteur T-DMB ses capacités.

Paramètres

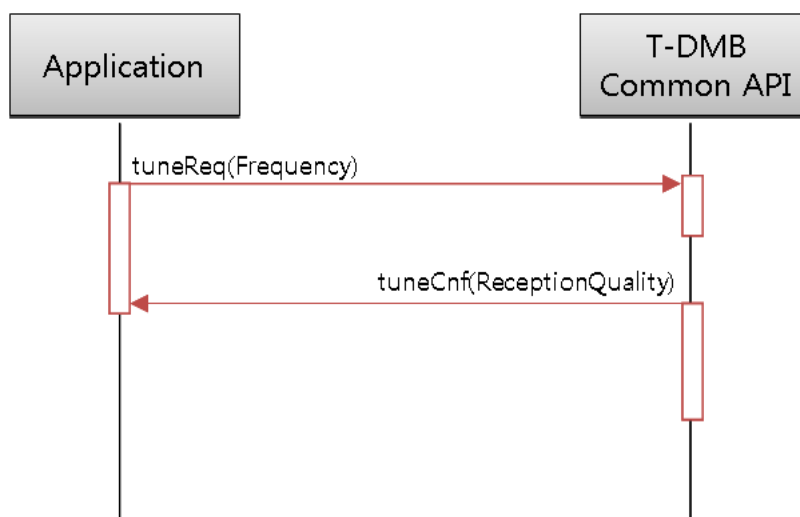
Néant

void capabilityCnf(ReceiverCapability)

Le récepteur T-DMB fournit ses capacités à l'API.

5.2.3 Syntonisation

La Figure 4 représente la syntonisation. Le récepteur T-DMB est syntonisé par appel tuneReq. Le récepteur se syntonise sur la fréquence demandée et répond ensuite au moyen de la confirmation tuneCnf. La confirmation contient des informations relatives à la qualité de la réception.



IEC 565/13

Légende

Anglais	Français
Application	Application
T-DMB Common API	API commune T-DMB

Figure 4 – Syntonisation

void tuneReq(int tuneFrequency,int transmissionMode)

La demande tuneReq initialise la commande Tune. La commande Tune règle directement une fréquence T-DMB spécifiée. Un récepteur T-DMB doit être syntonisé sur une fréquence T-DMB et synchronisé pour avoir accès aux services T-DMB. Un récepteur T-DMB syntonisé essaie automatiquement de se synchroniser sur un ensemble T-DMB. La commande Tune est utilisée pour sélectionner une fréquence T-DMB spécifiée. En fonction de la spécification pour le transmissionMode, on vérifie si un ensemble T-DMB peut être détecté. Si le récepteur T-DMB raccordé prend en charge la détection automatique, le réglage par défaut pour le transmissionMode peut être utilisé. Sinon, il doit spécifier les modes de transmission qu'il convient de soumettre à essai. Le résultat de la commande est délivré par la confirmation tuneCnf. Toutes les sélections de services audio et de données ou les sélections d'objets de données actuellement existantes sont automatiquement arrêtées avant que le récepteur T-DMB effectue la syntonisation.

Paramètres

- tuneFrequency - Ce paramètre spécifie la fréquence en hertz sur laquelle le récepteur T-DMB sera syntonisé.
- transmissionModes - Ce paramètre spécifie les modes de transmission essayés par un récepteur T-DMB pour des ensembles T-DMB. La valeur par défaut est T-DMBConstants.transmissionModeAutomatic ce qui signifie que le récepteur détecte automatiquement le mode de transmission. Le paramètre est un champ indicateur prenant en charge les indicateurs suivants, pouvant être spécifiés en même temps:
 - T-DMBConstants.transmissionModeAutomatic: Le mode de transmission est détecté automatiquement. Tous les autres indicateurs sont ignorés dans ce cas.
 - T-DMBConstants.transmissionMode1: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 1.
 - T-DMBConstants.transmissionMode2: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 2.
 - T-DMBConstants.transmissionMode3: À la fréquence spécifiée, on vérifie si un T-DMBEnsemble est envoyé dans le mode de transmission 3.

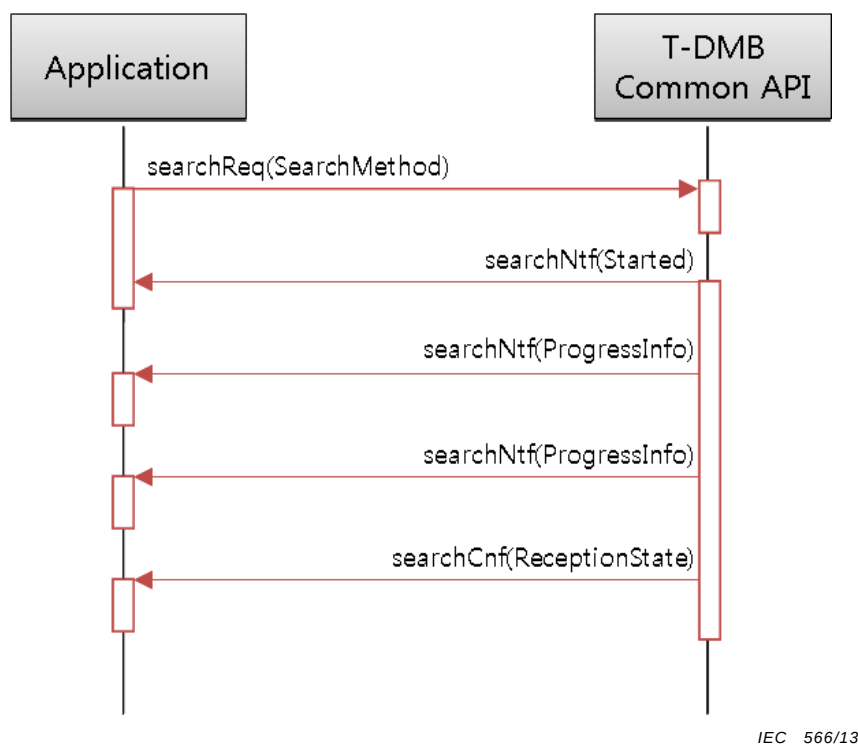
- T-DMBConstants.transmissionMode4: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 4.

void tuneCnf(TuneCnfEvent e)

La méthode TuneCnf finalise une commande Tune et est envoyée en réponse à un message TuneReq. Elle fournit des informations concernant la T-DMBfrequency actuellement sélectionnée et les conditions de réception. La commande Tune est utilisée pour sélectionner une T-DMBfrequency. La demande tuneReq initialise la commande Tune. Tunecnf finalise la commande Tune et fournit des informations concernant l'état de réception. Celles-ci comportent la fréquence sélectionnée, le mode de transmission détecté et l'état de synchronisation du récepteur.

5.2.4 Recherche

La Figure 5 représente la recherche. Pour rechercher un ensemble, l'application appelle searchReq. Les API communes T-DMB répondent par une notification de démarrage de la recherche. D'autres notifications sont envoyées entre-temps en fonction de la méthode de recherche (par exemple, si un pas de 16 kHz a été défini). La transaction se termine par une confirmation searchCnf contenant l'état résultant du processus de recherche.



IEC 566/13

Légende

Anglais	Français
Application	Application
T-DMB Common API	API commune T-DMB

Figure 5 – Recherche

```

void searchReq(
int searchMode,
int tables,
int startFrequency,
int stopFrequency,
int transmissionModes,
int notifications)
  
```

La demande searchReq initialise une commande Search. La commande Search recherche un ensemble T-DMB conformément à un mode de recherche spécifié. Après avoir réussi l'exécution de la commande Search, un ensemble T-DMB a été trouvé, on entre dans l'état Tuned et le récepteur T-DMB tente de se synchroniser automatiquement sur l'ensemble T-DMB trouvé. La commande Search est utilisée pour rechercher un ensemble T-DMB. La demande searchReq initialise la recherche et spécifie les fréquences et les modes de transmission à soumettre à essai. De plus, on peut spécifier les notifications obtenues par le client T-DMB pendant que la commande est exécutée. La recherche d'un ensemble peut nécessiter une durée substantielle allant seulement d'une seconde jusqu'à plusieurs minutes. Celle-ci dépend également du mode de recherche spécifié. Si les conditions de réception sont mauvaises, il est possible de ne détecter aucun ensemble T-DMB. Pour arrêter la recherche d'un ensemble T-DMB on peut utiliser la commande Tune qui syntonise le récepteur T-DMB sur une certaine fréquence indépendante des conditions de réception. Le démarrage de la recherche est indiqué par un événement SearchNtf avec le code d'état 'Started'. Dans ce cas, la machine d'état de l'état de syntonisation entre dans l'état recherche (voir Figure 5). Si l'état précédent a été syntonisé, toutes les sélections actuellement existantes de services ou d'objets sont automatiquement arrêtées. Pendant que la recherche est effectuée, plusieurs notifications délivrant des informations relatives à l'état actuel sont envoyées au client. La commande se termine par un événement SearchCnf.

Paramètres

- **searchMode** - Ce paramètre spécifie la façon dont le récepteur T-DMB recherche un ensemble T-DMB. Sa valeur par défaut est SearchModeAutomatic, ce qui signifie qu'il effectue la recherche conformément à une méthode par défaut. Le paramètre est un champ indicateur prenant en charge les indicateurs suivants, pouvant être spécifiés en même temps
 - T-DMBConstants.SearchModeAutomatic: méthode par défaut
 - T-DMBConstants.SearchMode16kHzSteps: La recherche s'effectue dans la plage de fréquences par pas de 16 kHz. Celle-ci est une recherche très conséquente, ce qui signifie que l'exécution de la commande peut prendre beaucoup de temps.
 - T-DMBConstants.SearchModeUp: La recherche s'effectue dans le sens des fréquences basses vers les fréquences hautes.
 - T-DMBConstants.SearchModeDown: La recherche s'effectue dans le sens des fréquences hautes vers les fréquences basses.
 - T-DMBConstants.SearchModeUseTables: La recherche est basée sur les tables de fréquences spécifiées.
 - T-DMBConstants.SearchModeUseFrequencyRange: La recherche est basée sur la plage de fréquences spécifiée.
 - T-DMBConstants.SearchModeContinuous: La recherche parcourt en boucle la plage de fréquences spécifiée jusqu'à avoir trouvé un ensemble T-DMB. Elle s'arrête par défaut après avoir contrôlé une fois la plage de fréquences spécifiée.
- **tables** - Ce paramètre spécifie les tables de fréquences utilisées par le récepteur pour rechercher les ensembles T-DMB. Le paramètre est un champ indicateur prenant en charge les indicateurs suivants, pouvant être spécifiés en même temps:
 - T-DMBConstants.searchCEPTFrequencyTableBandIII: L'essai des ensembles T-DMB est effectué sur les fréquences conformément au tableau des fréquences CEPT pour la Bande III.
 - T-DMBConstants.SearchCEPTFrequencyTableLBand: L'essai des ensembles T-DMB est effectué sur les fréquences conformément au tableau CEPT pour la Bande L.
 - T-DMBConstants.SearchCanadaFrequencyTableLBand: L'essai des ensembles T-DMB est effectué sur les fréquences conformément au tableau pour la Bande L canadienne.
- **transmissionModes** - Ce paramètre spécifie les modes de transmission essayés par un récepteur T-DMB pour des ensembles T-DMB. La valeur par défaut est T-DMBConstants.transmissionModeAutomatic ce qui signifie que le récepteur détecte

automatiquement le mode de transmission. Le paramètre est un champ indicateur prenant en charge les indicateurs suivants, pouvant être spécifiés en même temps:

- `transmissionModeAutomatic`: Le mode de transmission est détecté automatiquement. Tous les autres indicateurs sont ignorés.
- `transmissionMode1`: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 1.
- `transmissionMode2`: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 2.
- `transmissionMode3`: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 3.
- `transmissionMode4`: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 4.
- **notifications** - Ce paramètre spécifie le type de notification que le client souhaite obtenir pendant l'exécution de la commande `Seek`. Le paramètre est un champ indicateur prenant en charge les indicateurs suivants, pouvant être spécifiés en même temps:
 - `notificationsOff`: Aucune notification intermédiaire n'est envoyée. Seule une notification `SearchNtf` indiquant le démarrage de la recherche est envoyée.
 - `notifications16kHzSteps`: Une notification est envoyée à chaque pas de 16 kHz. Celle-ci n'est utilisée que si la recherche par pas de 16 kHz est spécifiée comme mode de recherche.
 - `notificationsTableEntry`: Une notification est envoyée à chaque fréquence d'entrée de la table. Ceci est la valeur par défaut.

void searchCnf(SearchCnfEvent e)

La méthode `searchCnf` finalise une commande `Search` et fournit des informations relatives à l'état de la commande, à la fréquence T-DMB actuellement sélectionnée et aux conditions de réception actuelles. La commande `Search` est utilisée pour rechercher un ensemble T-DMB conformément à un mode de recherche spécifié. La recherche d'un ensemble T-DMB peut prendre beaucoup de temps. Le démarrage de la recherche est indiqué par un message `searchNtf 'Started'`. D'autres messages `searchNtf` informent un client T-DMB de l'avancement de la recherche. Celle-ci est finalisée par la fourniture du message `searchCnf`. Celui-ci informe de l'état de la commande, de la fréquence sélectionnée et de l'état de synchronisation. Aucun autre message `searchNtf` ne sera délivré après le message `searchCnf`.

void searchNtf(SearchNtfEvent e)

L'événement `SearchNtf` est envoyé après avoir démarré une recherche `searchReq` d'ensemble T-DMB. Il informe du démarrage de la recherche et de l'avancement de la recherche. Un événement `SearchCnf` finalise une commande `Search`. Aucun autre événement `SearchNtf` n'est envoyé après avoir envoyé un événement `SearchCnf`. L'événement `SearchNtf` est envoyé après avoir démarré la recherche d'un ensemble T-DMB et pendant que la recherche est en cours afin de fournir des informations sur l'état actuel de la recherche. La notification `'Started'` est envoyée dans tous les cas. Les notifications d'avancement ne sont envoyées que si des notifications ont été demandées avec le message `SearchReq` associé. Aucune autre notification ne sera envoyée après avoir délivré un message `SearchCnf`.

5.2.5 Balayage

La Figure 6 représente le balayage. Le balayage signifie la recherche d'ensembles sur une plage spécifiée. Il ressemble essentiellement à la recherche, à l'exception du fait que le processus de balayage recherche tous les ensembles dans la plage. Lorsque la commande a été délivrée, une notification est envoyée, après avoir démarré le balayage. D'autres notifications sont envoyées pendant le balayage, informant de son avancement. Lorsque le balayage est terminé, une confirmation est envoyée, contenant des informations relatives au balayage et à l'état du récepteur.

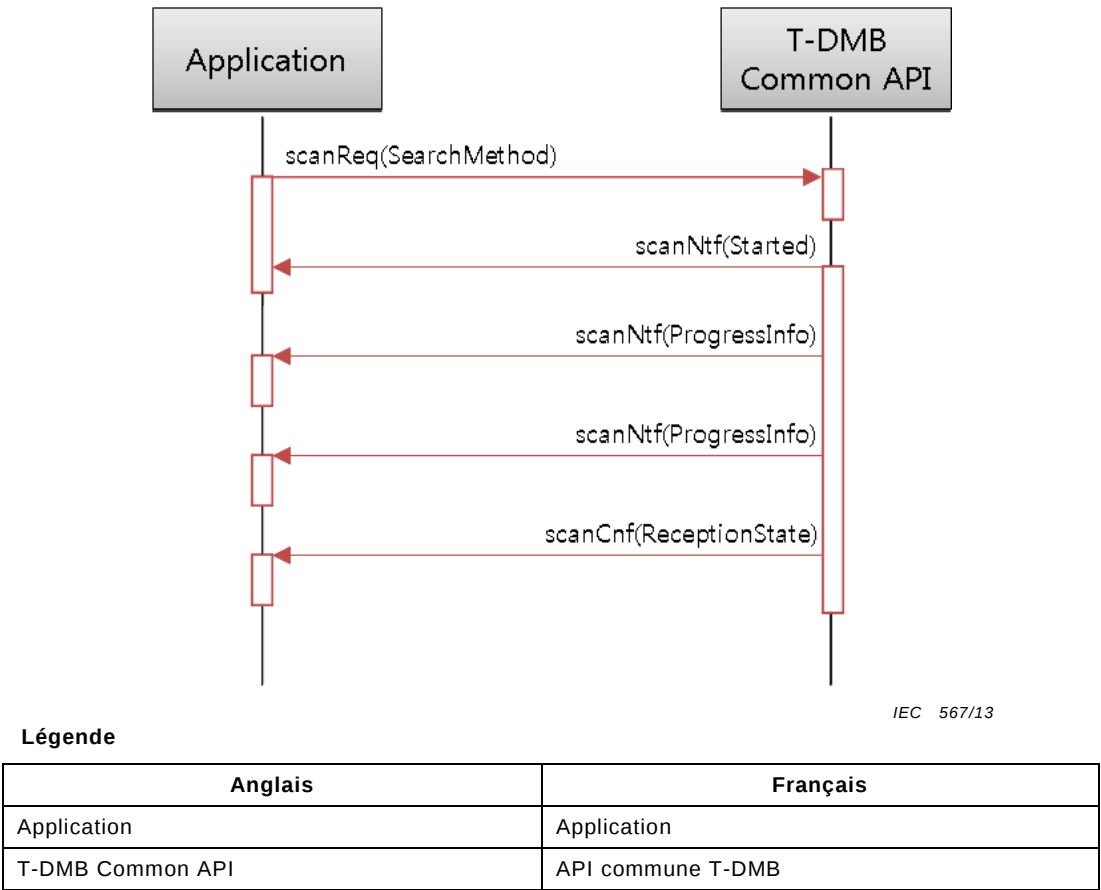


Figure 6 – Balayage

```
void scanReq(
int searchMode,
int tables,
int startFrequency,
int stopFrequency,
int tansmissionModes,
int notifications)
```

La demande ScanReq initialise une commande Scan. La commande Scan est utilisée pour effectuer une recherche de tous les ensembles T-DMB dans une plage de fréquences spécifiée. En fonction de la plage de fréquences et du mode de recherche, cette opération peut nécessiter une durée substantielle, allant seulement d'une seconde jusqu'à plusieurs minutes. La commande démarre par la demande ScanReq et se termine par la confirmation ScanCnf. Entre-temps, des notifications ScanNtf sont envoyées pour informer de l'état actuel du balayage si des notifications sont demandées. Dans le cas d'une recherche des fréquences inférieures vers des fréquences supérieures (searchMode= T-DMBConstants.searchModeUp).

Il n'est pas autorisé que la valeur de startFrequency soit plus grande que la valeur de stopFrequency. Dans le cas d'une recherche des fréquences supérieures vers les fréquences inférieures (searchMode= T-DMBConstants.searchModeDown), il n'est pas autorisé que la valeur de startFrequency soit inférieure à la valeur de stopFrequency.

Paramètres

- **searchMode** - Ce paramètre spécifie la façon dont le récepteur T-DMB recherche un ensemble T-DMB. Sa valeur par défaut est T-DMBConstants.searchModeAutomatic, ce qui signifie qu'il effectue la recherche conformément à une méthode par défaut. Le paramètre est un champ indicateur prenant en charge les indicateurs suivants, pouvant être spécifiés en même temps:

- T-DMBConstants.SearchModeAutomatic: méthode par défaut
- T-DMBConstants.SearchMode16kHzSteps: La recherche s'effectue dans la plage de fréquences par pas de 16 kHz.
- T-DMBConstants.SearchModeUp: La recherche s'effectue dans le sens des fréquences basses vers les fréquences hautes.
- T-DMBConstants.SearchModeDown: La recherche s'effectue dans le sens des fréquences hautes vers les fréquences basses.
- T-DMBConstants.SearchModeUseTables: La recherche est basée sur les tables de fréquences spécifiées.
- T-DMBConstants.SearchModeUseFrequencyRange: La recherche est basée sur la plage de fréquences spécifiée.
- T-DMBConstants.SearchModeContinuous: La recherche parcourt en boucle la plage de fréquences spécifiée jusqu'à avoir trouvé un ensemble T-DMB. Elle s'arrête par défaut après avoir contrôlé une fois la plage de fréquences spécifiée.
- **tables** - Ce paramètre spécifie les tables de fréquences utilisées par le récepteur pour rechercher les ensembles T-DMB. Le paramètre est un champ indicateur prenant en charge les indicateurs suivants, pouvant être spécifiés en même temps:
 - T-DMBConstants.searchCEPTFrequencyTableBandIII: L'essai des ensembles T-DMB est effectué sur les fréquences conformément au tableau des fréquences CEPT pour la Bande III.
 - T-DMBConstants.SearchCEPTFrequencyTableLBand: L'essai des ensembles T-DMB est effectué sur les fréquences conformément au tableau CEPT pour la Bande L.
 - T-DMBConstants.SearchCanadaFrequencyTableLBand: L'essai des ensembles T-DMB est effectué sur les fréquences conformément au tableau pour la Bande L canadienne.
- **startFrequency** - Ce paramètre spécifie la fréquence de démarrage à laquelle le récepteur T-DMB démarre sa recherche des ensembles T-DMB.
- **stopFrequency** - Ce paramètre spécifie la fréquence d'arrêt à laquelle le récepteur T-DMB arrête sa recherche des ensembles T-DMB.
- **transmissionModes** - Ce paramètre spécifie les modes de transmission dans lesquels il convient qu'un récepteur T-DMB recherche des ensembles T-DMB. La valeur par défaut est T-DMBConstants.transmissionModeAutomatic ce qui signifie que le récepteur détecte automatiquement le mode de transmission. Le paramètre est un champ indicateur prenant en charge les indicateurs suivants, pouvant être spécifiés en même temps:
 - T-DMBConstants.transmissionModeAutomatic: Le mode de transmission est détecté automatiquement.
 - T-DMBConstants.transmissionMode1: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 1.
 - T-DMBConstants.transmissionMode2: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 2.
 - T-DMBConstants.transmissionMode3: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 3.
 - T-DMBConstants.transmissionMode4: À la fréquence spécifiée, on vérifie si un ensemble T-DMB est envoyé dans le mode de transmission 4.
 - **notifications** - Ce paramètre spécifie le type de notification souhaitée par l'application pendant l'exécution de la commande Seek. Le paramètre est un champ indicateur prenant en charge les indicateurs suivants, pouvant être spécifiés en même temps:
 - T-DMBConstants.notificationsOff: Aucune notification n'est envoyée.
 - T-DMBConstants.notifications16kHzSteps: Une notification est envoyée à chaque pas de 16 kHz.
 - T-DMBConstants.notificationsTableEntry: Une notification est envoyée à chaque fréquence d'entrée de la table. Ceci est la valeur par défaut.

void scanCnf(ScanCnfEvent e)

Le message ScanCnf finalise une commande Scan. Il informe du résultat du balayage et de l'état actuel de syntonisation. La commande Scan est utilisée pour effectuer une recherche de tous les ensembles T-DMB dans une plage de fréquences spécifiée. En fonction de la plage de fréquences et du mode de recherche, cette opération peut nécessiter une durée substantielle, allant seulement d'une seconde jusqu'à plusieurs minutes. La commande démarre par le message ScanReq et se termine par le message ScanCnf. Entre-temps, des messages ScanNtf sont envoyés pour informer de l'état actuel de la recherche si des notifications sont demandées.

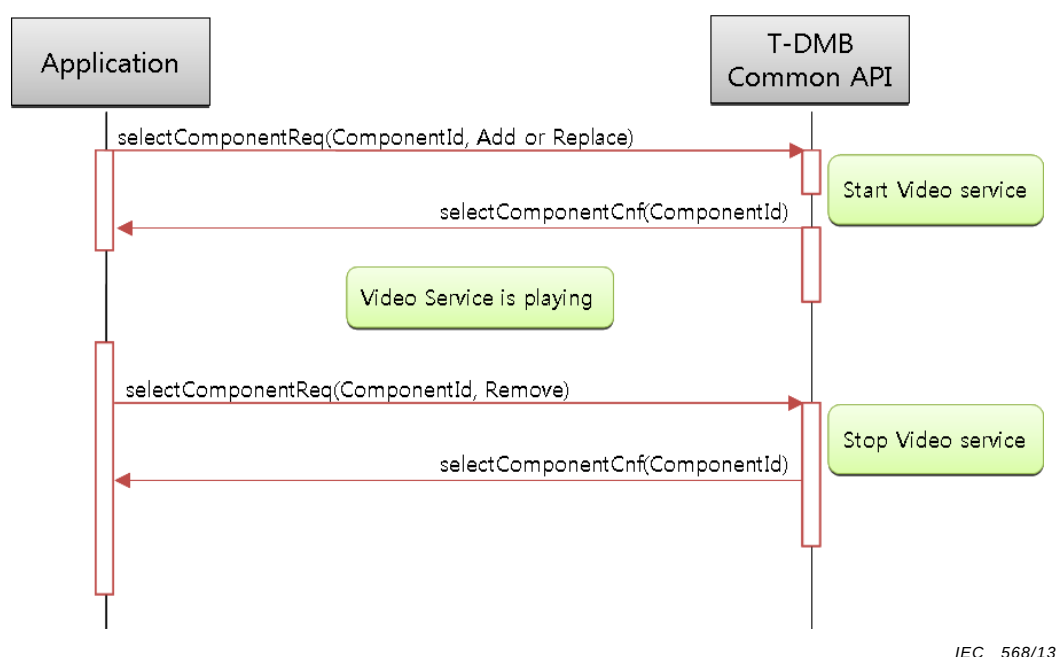
Le message ScanCnf indique que la commande de balayage est terminée et informe de l'état actuel de syntonisation. À la suite de l'exécution de la commande de balayage, des informations sont introduites dans la base de données d'informations de service. Si un abonnement SI est en cours, plusieurs messages SINtf sont délivrés à l'application connectée.

void scanNtf(ScanNtfEvent e)

Le message ScanNtf est envoyé après démarrage par le message ScanReq de la recherche de tous les ensembles T-DMB disponibles dans une plage de fréquences spécifiée. Le message ScanNtf donne des informations concernant l'état actuel de la recherche pour tous les ensembles T-DMB dans une plage de fréquences spécifiée. Il est délivré à l'application connectée après démarrage de la recherche par le message ScanReq et demande des notifications. Aucune autre notification ne sera envoyée après avoir délivré un message ScanCnf message.

5.2.6 Sélection d'un service T-DMB

La Figure 7 représente la sélection d'un service T-DMB. La commande SelectComponent démarre un service audio/vidéo. L'application appelle selectComponentReq qui transmet l'identifiant du composant audio/vidéo. Les API communes T-DMB démarrent le service audio/vidéo et renvoient une confirmation. Pour arrêter ce service audio/vidéo, l'application appelle de nouveau selectComponentReq en spécifiant alors que le composant doit être retiré. Lorsque les API communes T-DMB répondent par une confirmation, le service audio/vidéo a été arrêté.



Légende

Anglais	Français
Application	Application
T-DMB Common API	API commune T-DMB
Start Video service	Démarrage service vidéo
Video service is playing	Service vidéo en cours
Stop Video service	Arrêt service vidéo

Figure 7 – Sélection d'un service T-DMB

**void selectComponentReq(
ComponentId id,
int selectionMode)**

La demande selectComponentReq initialise la commande SelectComponent. La commande SelectComponent démarre ou arrête une application délivrée dans un composant T-DMB. La commande SelectComponent permet de démarrer ou d'arrêter des applications délivrées dans des composants T-DMB. Généralement, plusieurs composants du même ensemble T-DMB peuvent être sélectionnés simultanément. Il est possible de sélectionner un composant audio, tous les composants de données associés au programme du composant audio sélectionné et plusieurs composants de données indépendants en même temps. La sélection d'un composant est demandée par la demande selectComponentReq et elle est confirmée par la confirmation selectComponentCnf. Il est possible de retirer un composant d'un ensemble T-DMB, ce qui signifie qu'il n'est plus diffusé et n'est donc plus disponible. Ceci est indiqué par un appel SINTf et signifie que la sélection est automatiquement retirée. Si la sélection d'un composant est retirée, toutes les sélections d'objets existants appartenant au composant sont enlevées.

Si l'application de l'utilisateur est un diaporama ou une étiquette dynamique, ses objets sont délivrés automatiquement (en utilisant les notifications objectNtf) après avoir envoyé la confirmation SelectComponent. Si le composant sélectionné est un service audio, ses services de données PAD deviennent également disponibles. Ceci signifie que des informations de service sont générées pour tous les services PAD et qu'elles peuvent être sélectionnées. Si la sélection du service audio est arrêtée, tous les services PAD sont également arrêtés et ils ne sont plus disponibles. Si le composant n'est pas l'ensemble actuel, le fait qu'il soit sélectionné dépend néanmoins de la mise en œuvre.

Paramètres

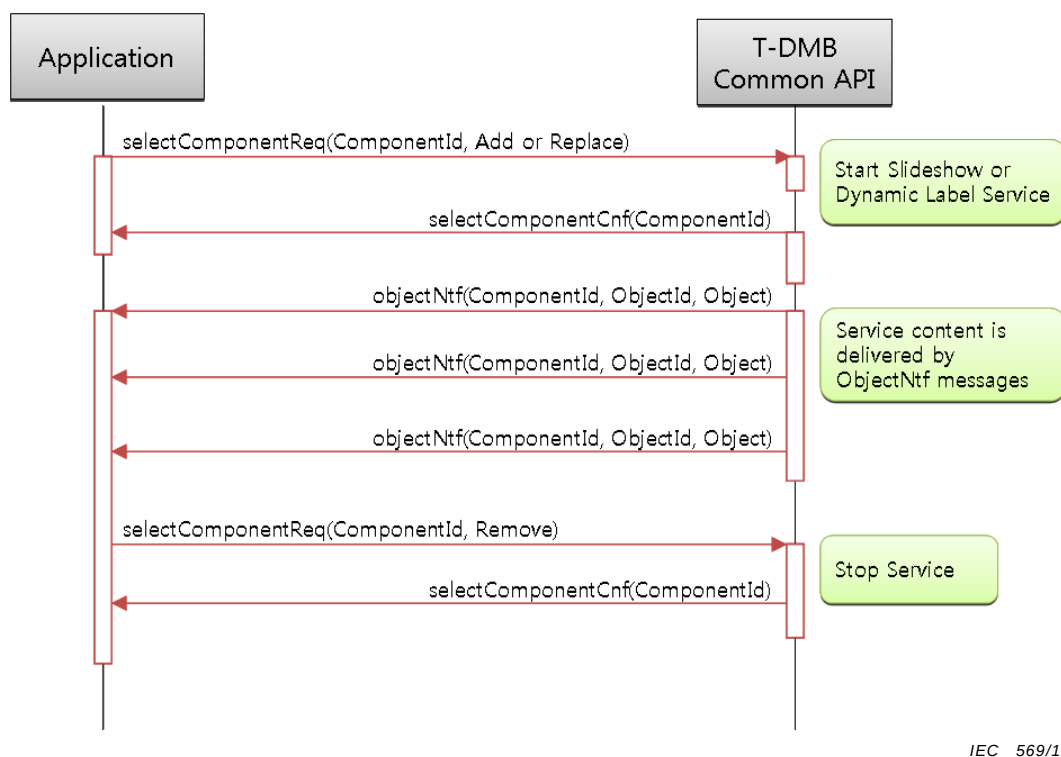
- **id** - Ce paramètre est un pointeur sur l'identifiant du composant T-DMB qui doit être sélectionné. S'il convient de retirer toutes les sélections de composants (mettre `selectionMode` à `T-DMBConstants.selectionModeRemoveAll`) ce paramètre est ignoré et il convient de l'annuler.
- **selectionMode** - Ce paramètre spécifie le mode de sélection pour le composant. Les indicateurs suivants sont pris en charge:
 - `T-DMBConstants.selectionModeReplace`: Tous les composants du même type actuellement sélectionnés sont arrêtés et le composant spécifié doit démarrer. Même type signifie qu'un composant audio remplace tout autre composant audio sélectionné, un composant de données remplace tous les autres composants de données indépendants sélectionnés et un composant de données associé à un programme remplace tous les autres composants de données associés à un programme sélectionné.
 - `T-DMBConstants.selectionModeAdd`: L'application délivrée par le composant spécifié est sur le point de démarrer. Les autres composants sélectionnés ne sont pas affectés.
 - `T-DMBConstants.selectionModeRemove`: La sélection du composant spécifié est arrêtée.
 - `T-DMBConstants.selectionModeRemoveAll`: Toutes les sélections de composants existantes sont retirées. Dans ce cas, annuler `serviceId`.

`void selectComponentCnf(SelectComponentCnfEvent e)`

La confirmation `SelectComponentCnf` finalise la commande `SelectComponent`. Elle informe de l'état de la commande et de l'état de sélection du composant spécifié. La commande `SelectComponent` permet de démarrer ou d'arrêter des applications délivrées dans des composants T-DMB. Généralement, plusieurs composants du même ensemble T-DMB peuvent être sélectionnés simultanément. Il est possible de sélectionner un composant audio, tous les composants de données associés au programme du composant audio sélectionné et plusieurs composants de données indépendants en même temps. La sélection d'un composant est demandée par le message `selectComponentReq` et est confirmée par un appel `selectComponentCnf`. Il est possible de retirer un composant d'un ensemble T-DMB, ce qui signifie qu'il n'est plus diffusé et n'est donc plus disponible. Ceci est indiqué par un appel `SINtf` et signifie que la sélection est automatiquement retirée.

5.2.7 Sélection d'un diaporama ou d'un service d'étiquettes dynamiques

La Figure 8 représente la sélection d'un diaporama ou d'un service d'étiquettes dynamiques. Une application sélectionne un diaporama ou un service d'étiquettes dynamiques au moyen de la commande `SelectComponent`. Lorsque la demande `selectComponentReq` avec l'identifiant de service respectif est délivrée, le service démarre et une confirmation est renvoyée. L'application reçoit ensuite des notifications `objectNtf` contenant les objets du service. Pour arrêter le service, `selectComponentReq` est de nouveau appelé, mettant `selectionMode` à `selectionModeRemove`. Le retrait du service est confirmé.



Légende

Anglais	Français
Application	Application
T-DMB Common API	API commune T-DMB
Start Slideshow or Dynamic Label Service	Démarrage diaporama ou service d'étiquettes dynamiques
Service content is delivered by ObjectNtf messages	Le contenu du service est délivré par des messages ObjectNtf
Stop Service	Arrêt du service

Figure 8 – Sélection d'un diaporama ou d'un service d'étiquettes dynamiques

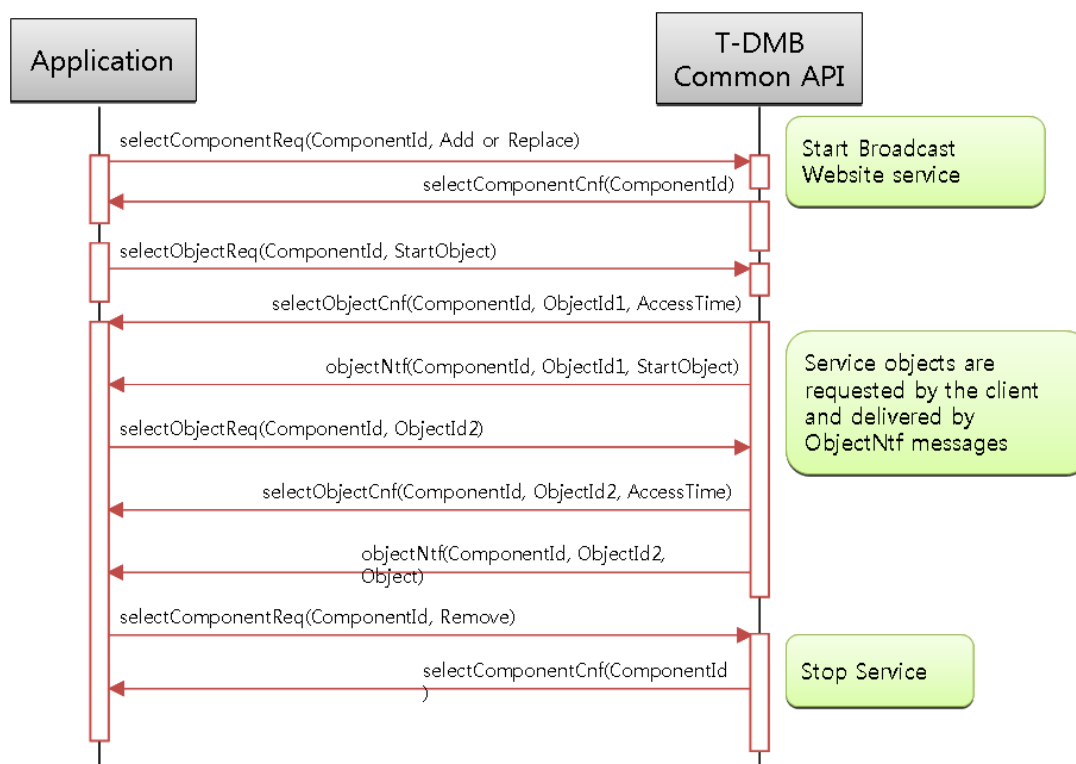
void objectNtf(ObjectNtfEvent e)

La méthode `objectNtf` est appelée en conséquence de la sélection d'objets d'un composant de données en utilisant la commande `SelectObject`. Elle délivre à un client T-DMB un objet partiellement sélectionné ou complet. On utilise `objectNtf` pour délivrer un objet sélectionné au client T-DMB connecté. En fonction du mode de demande, l'objet est délivré une seule fois ou plusieurs fois dans le cas de mises à jour. Si l'objet ne peut pas être délivré à temps comme indiqué par un appel à `selectObjectCnf`, `objectNtf` informe alors du retard. Si la transmission d'un objet sélectionné est arrêtée, `objectNtf` informe de la fin de la transmission de l'objet et de la sélection de l'objet. Il est possible de retirer un composant T-DMB d'un ensemble T-DMB. Ceci est indiqué par un appel à `siNtf`. Dans ce cas également, les objets sélectionnés du composant ne sont plus sélectionnés. Aucun message de fin n'est envoyé pour les sélections d'objets terminées résultant de la fin d'un composant.

5.2.8 Sélection d'un service de site Web à large diffusion

La Figure 9 représente la sélection d'un service de site Web à large diffusion. Pour exécuter un service de site Web à large diffusion, le composant doit être sélectionné. Ceci est réalisé en appelant `selectComponentReq` avec l'identifiant de service respectif. Le démarrage du service est confirmé par les API communes T-DMB. Les objets réels du service sont récupérés au moyen de la commande `selectObject`. L'objet de démarrage est généralement demandé en premier. Pour cela, une demande `selectObjectReq` est délivrée avec l'identifiant de service du composant et l'identifiant d'objet de l'objet de démarrage. Les API communes T-

DMB renvoie une confirmation incluant le temps d'accès probable. L'objet réel est reçu avec une notification objectNtf. Tous les autres objets du service sont demandés et délivrés de manière similaire. Le service est arrêté en appelant selectComponentReq, qui spécifie le retrait du service. Noter que la commande SelectComponent peut être utilisée pour améliorer le temps d'accès du temps désiré (par exemple, en particulier en mettant en antémemoire les objets du service).



IEC 570/13

Légende

Anglais	Français
Application	Application
T-DMB Common API	API commune T-DMB
Start Broadcast Website service	Démarrage du service de site Web à large diffusion
Service objects are requested by the client and delivered by ObjectNtf messages	Les objets du service sont demandés par le client et délivrés par les messages ObjectNtf
Stop Service	Arrêt du service

Figure 9 – Sélection d'un service de site Web à large diffusion

```

void selectObjectReq(
ComponentId id,
ObjectId objectId,
int requestMode,
boolean replaceSelections,
int deliveryMode,
int cacheHint)
    
```

La demande selectObjectReq initialise la commande SelectObject. La commande SelectObject sélectionne un objet d'un composant T-DMB sélectionné. Celle-ci comporte la demande d'un objet d'un composant de données, la fourniture après réception et la notification des mises à jour tant que l'objet est sélectionné. La sélection signifie qu'il est demandé qu'il soit fourni et si on le souhaite, des mises à jour de l'objet sont également

délivrées. De plus, il est possible de fournir certaines indications pour la mise en antémémoire. Plusieurs objets peuvent être sélectionnés simultanément, pouvant également provenir de plusieurs composants. La sélection d'un objet est demandée par la demande `selectObjectReq` et elle est confirmée par la confirmation `selectObjectCnf`. L'objet est délivré en utilisant la méthode `objectNtf`. Celle-ci comporte la première fourniture et toutes les mises à jour. Outre le démarrage ou l'arrêt d'une sélection, il est possible de supprimer toutes les autres sélections appartenant au même composant en mettant à vrai le paramètre `replaceSelections`. Il est possible de retirer un composant d'un ensemble T-DMB. Ceci est indiqué par un appel `serviceInfoNtf`. Dans ce cas également, les objets sélectionnés du service ne sont plus sélectionnés. Il est possible de retirer un objet du service en cours d'émission. Ceci est indiqué par un appel `objectNtf`. Dans ce cas, les sélections pour cet objet sont automatiquement désactivées.

Actuellement, la sélection d'objets n'a de sens qu'avec des applications du type `BroadcastWebSite`. Les objets d'applications telles que `Slideshows` ou `Dynamic Label` sont délivrés automatiquement par les appels `objectNtf`.

Paramètres

- **id** - Ce paramètre identifie le composant sélectionné auquel appartient l'objet.
- **objectId** - Ce paramètre identifie l'objet qui doit être sélectionné.
- **selectionMode** - Ce paramètre spécifie le mode de sélection de l'objet. Les valeurs suivantes sont prises en charge:
 - `T-DMBConstants.requestModeOff`: Celle-ci est utilisée pour arrêter la sélection d'objets qui sont demandés avec le mode de demande `T-DMBConstants.requestModeUpdate`. Elle n'est pas nécessaire pour les objets qui sont demandés avec l'indicateur `T-DMBConstants.requestModeOnce`, sauf lorsque un `SelectObjectReq` est en attente et que la fourniture n'est plus désirée.
 - `T-DMBConstants.requestModeOnce`: L'objet est demandé pour une seule fourniture. Après la première réception du canal de diffusion large, l'objet est délivré au client T-DMB connecté. Le client n'est pas averti des nouvelles versions.
 - `T-DMBConstants.requestModeUpdate`: L'objet est demandé pour fourniture de mise à jour. Après la première réception du canal de diffusion large, l'objet est délivré au client connecté. De plus, chaque nouvelle version de l'objet est délivrée.
- **replaceSelections** - Ce paramètre spécifie si toutes les sélections actuelles d'objets appartenant au composant identifié par `serviceId` sont remplacées par cette sélection. Si ce paramètre est mis à vrai, toutes les sélections sont alors retirées. Si ce paramètre est mis à faux, les sélections existantes restent alors inchangées.
- **deliveryMode** - Ce paramètre spécifie le mode de fourniture de l'objet. Les valeurs suivantes sont prises en charge:
 - `T-DMBConstants.deliveryModeComplete`: Seul l'objet complet est délivré au client T-DMB.
 - `T-DMBConstants.deliveryModePartial`: L'objet peut être délivré par parties.
- **cacheHint** - Ce paramètre spécifie une indication pour la mise en antémémoire de l'objet sélectionné.

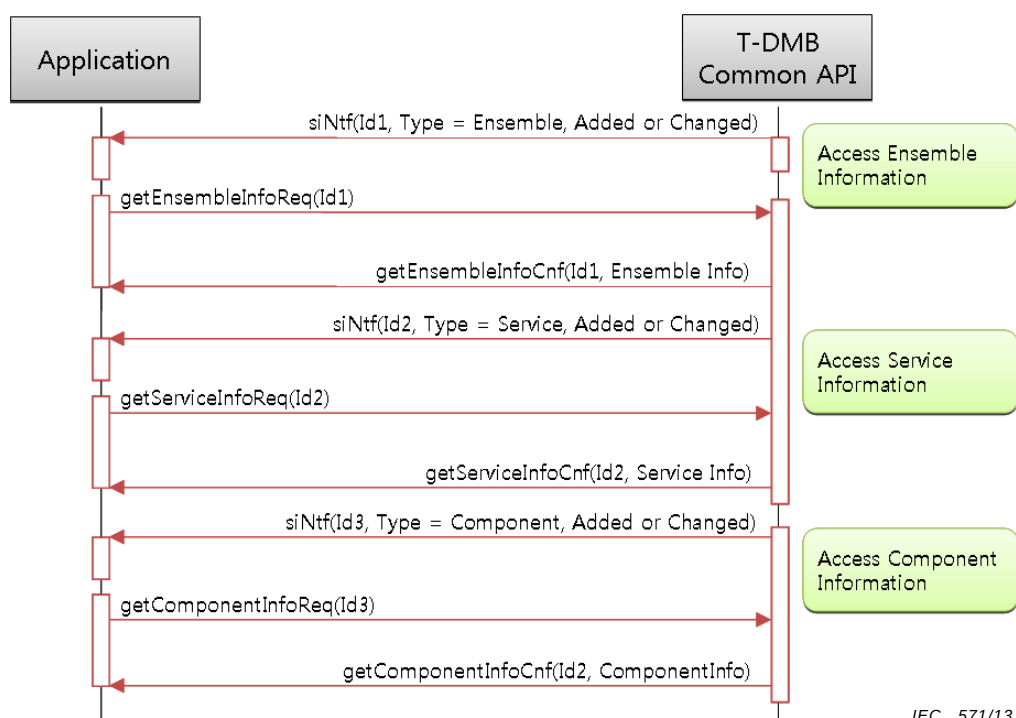
`void selectObjectCnf(SelectObjectCnfEvent e)`

La méthode `SelectObjectCnf` finalise la commande `SelectObject`. La commande `SelectObject` sélectionne un objet d'un composant T-DMB sélectionné. Celle-ci comporte la demande d'un objet d'un composant de données, la fourniture après réception et la notification des mises à jour tant que l'objet est sélectionné. La sélection signifie qu'il est demandé qu'il soit fourni et si on le souhaite, des mises à jour de l'objet sont également délivrées. De plus, il est possible de fournir certaines indications pour la mise en antémémoire. Plusieurs objets peuvent être sélectionnés simultanément, pouvant également provenir de plusieurs composants. La sélection d'un objet est demandée par `selectObjectReq` et elle est confirmée par l'appel `selectObjectCnf`. L'objet est délivré en utilisant `objectNtf`. Celui-ci comporte la première

fourniture et toutes les mises à jour. Outre le démarrage ou l'arrêt d'une sélection, il est possible de supprimer toutes les autres sélections appartenant au même composant en mettant à vrai le paramètre replaceSelections. Il est possible de retirer un composant d'un ensemble T-DMB. Dans ce cas également, les objets sélectionnés du service ne sont plus sélectionnés. Il est possible de retirer un objet du service en cours d'émission. Ceci est indiqué en appelant objectNtf. Dans ce cas, les sélections pour cet objet sont automatiquement désactivées. Actuellement, la sélection d'objets n'a de sens qu'avec des applications du type BroadcastWebSite. Les objets d'applications telles que Slideshows ou Dynamic Label sont délivrés automatiquement en utilisant objectNtf.

5.2.9 Informations de service Get T-DMB

La Figure 10 représente les informations de service Get T-DMB. L'application peut également utiliser la commande ServiceInfo pour récupérer les objets d'informations de service T-DMB respectifs. Elle doit spécifier l'identifiant de service dans la demande siReq. La confirmation contient alors l'objet demandé.



Légende

Anglais	Français
Aplication	Application
T-DMB Common API	API commune T-DMB
Access Ensemble Information	Accès aux informations d'ensemble
Access Service Information	Accès aux informations de service
Access Component Information	Accès aux informations de composant

Figure 10 – Informations de service Get T-DMB

void getEnsembleInfoReq(EnsembleId id)

La méthode getEnsembleInfoReq initialise une commande GetEnsembleInfo. La commande GetEnsembleInfo demande des informations concernant l'ensemble T-DMB spécifié. La commande GetEnsembleInfo fournit à un client T-DMB des informations concernant un ensemble T-DMB spécifié, par exemple étiquette, n° de services, et ainsi de suite. La commande est initialisée par une demande getEnsembleInfoReq et se termine par une confirmation getEnsembleInfoCnf.

Paramètres

- **id** - Ce paramètre est un descripteur identifiant l'ensemble T-DMB.

void getServiceInfoReq(ServiceId id)

La demande getServiceInfoReq initialise une commande GetServiceInfo. La commande GetServiceInfo demande des informations concernant un service T-DMB spécifié. La commande GetServiceInfo fournit à un client T-DMB des informations concernant un service T-DMB spécifié, par exemple étiquette, n° de composants, et ainsi de suite. La commande est initialisée par une demande getServiceInfoReq et se termine par une confirmation getServiceInfoCnf.

Paramètres

- **id** - Ce paramètre est un descripteur identifiant le service T-DMB.

void getComponentInfoReq(ComponentId id)

La demande getComponentInfoReq initialise une commande GetComponentInfo. La commande GetComponentInfo demande des informations concernant un composant T-DMB spécifié. La commande GetComponentInfo fournit à un client T-DMB des informations concernant un composant T-DMB spécifié, par exemple étiquette, langue, et ainsi de suite. La commande est initialisée en appelant getComponentInfoReq et se termine par un appel à getComponentInfoCnf.

Paramètres

- **id** - Ce paramètre est un descripteur identifiant le composant T-DMB.

void getEnsembleInfoCnf(GetEnsembleInfoCnfEvent e)

La méthode GetEnsembleInfoCnf finalise la commande GetEnsembleInfo et délivre des informations concernant un ensemble T-DMB demandé à un client T-DMB. La commande GetEnsembleInfo fournit à un client T-DMB des informations concernant un ensemble T-DMB spécifié, par exemple étiquette, No of Services, et ainsi de suite. La commande est initialisée par une demande getEnsembleInfoReq et se termine par un appel getEnsembleInfoCnf.

void getServiceInfoCnf(GetServiceInfoCnfEvent e)

Un appel à la méthode getServiceInfoCnf finalise la commande GetServiceInfo et délivre des informations concernant un service T-DMB demandé à un client T-DMB. La commande GetServiceInfo fournit à un client T-DMB des informations concernant un service T-DMB spécifié, par exemple étiquette, n° de services, et ainsi de suite. La commande est initialisée par un message getServiceInfoReq et se termine par un message getServiceInfoCnf.

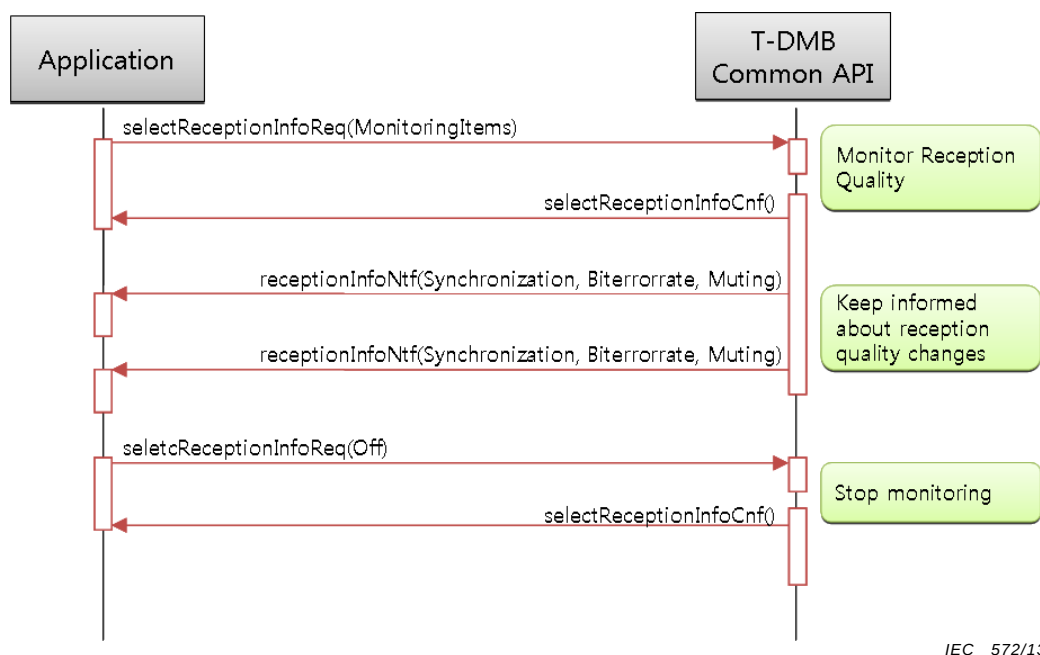
void getComponentInfoCnf(GetComponentInfoCnfEvent e)

Le message GetComponentInfoCnf finalise la commande GetComponentInfo et délivre des informations concernant un composant T-DMB demandé à un client T-DMB. La commande GetComponentInfo fournit à un client T-DMB des informations concernant un composant T-DMB spécifié, par exemple étiquette, langue, et ainsi de suite. La commande est initialisée par une demande GetComponentInfoReq et se termine par un appel au message GetComponentInfoCnf.

5.2.10 Surveillance de la qualité de réception

La Figure 11 représente la surveillance de la qualité de réception. La qualité de réception peut être surveillée en utilisant la commande SelectReceptionInfo. L'application doit effectuer une demande selectReceptionInfoReq spécifiant les paramètres qui doivent être surveillés. Elle reçoit ensuite des notifications receptionInfoNtf tant que la surveillance n'est pas arrêtée

(selectionReceptionInfoReq (Off)).



Légende

Anglais	Français
Application	Application
T-DMB Common API	API commune T-DMB
Monitor Reception Quality	Surveillance de la qualité de réception
Keep informed about reception quality changes	Maintien informé des changements de qualité de la réception
Stop monitoring	Arrêt de la surveillance

Figure 11 – Surveillance de la qualité de réception

**void selectReceptionInfoReq(
boolean synchronizationNotification,
boolean bitErrorRateNotifcations,
boolean muteStateNotifications,
boolean requestOnce)**

La demande selectReceptionInfoReq initialise la commande SelectReceptionInfo. La commande SelectReceptionInfo démarre, arrête ou modifie l'abonnement aux notifications de changement d'état concernant les conditions de réception. Il est possible de surveiller la synchronisation, le taux d'erreur sur les bits et l'inhibition du décodeur audio. La commande SelectReceptionInfo permet à un client T-DMB de s'abonner aux notifications de changement d'état concernant les conditions de réception en termes de synchronisation, taux d'erreur sur les bits et inhibition du décodeur audio. L'abonnement est demandé par la demande selectReceptionInfoReq et il est confirmé par la confirmation selectReceptionInfoCnf. Le niveau de l'abonnement peut être modifié par une autre commande SelectReceptionInfo. Celle-ci comporte l'arrêt de l'abonnement. Après avoir réussi à s'abonner, le client T-DMB appelant reçoit des notifications ReceptionInfoNtf lorsque des changements d'état se produisent.

Paramètres

- **synchronizationNotification** - Ce paramètre spécifie si le client appelant est averti des changements d'état concernant la synchronisation du signal T-DMB. Si le paramètre est

mis à vrai (par défaut), des notifications sont fournies, s'il est mis à faux, aucune notification n'est fournie.

- **bitErrorRateNotifications** - Ce paramètre spécifie si le client appelant est averti des changements d'état concernant le taux d'erreur sur les bits. Si le paramètre est mis à vrai (par défaut), des notifications sont fournies, s'il est mis à faux, aucune notification n'est fournie.
- **muteStateNotifications** - Ce paramètre spécifie si le client appelant est averti des changements d'état concernant l'état d'inhibition du décodeur audio. Si le paramètre est mis à vrai (par défaut), des notifications sont fournies, s'il est mis à faux, aucune notification n'est fournie.
- **requestOnce** - Ce paramètre spécifie si les informations de conditions de réception ne sont désirées qu'une seule fois. Dans ce cas, les conditions de réception sont détectées une fois et le client T-DMB est informé par un unique appel `receptionInfoNtf`.

void selectReceptionInfoCnf(SelectReceptionInfoCnfEvent e)

La méthode `selectReceptionInfoCnf` finalise la commande `SelectReceptionInfo`. Elle informe de l'état de la commande et du niveau d'abonnement actuel. La méthode `SelectReceptionInfo` permet à un client T-DMB de s'abonner aux notifications de changement d'état concernant les conditions de réception en termes de synchronisation, taux d'erreur sur les bits et inhibition du décodeur audio. L'abonnement est demandé par `selectReceptionInfoReq` et il est confirmé par `selectReceptionInfoCnf`. Le niveau de l'abonnement peut être modifié par une autre commande `SelectReceptionInfo`. Celle-ci comporte l'arrêt de l'abonnement. Après avoir réussi à s'abonner, le client T-DMB appelant reçoit des appels `ReceptionInfoNtf` lorsque des changements d'états se produisent.

void receptionInfoNtf(ReceptionInfoNtfEvent e)

La méthode `receptionInfoNtf` est appelée à la suite de l'abonnement à des changements d'état de synchronisation, taux d'erreur sur les bits et inhibition du décodeur audio. `receptionInfoNtf` indique que l'état de synchronisation, le taux d'erreur sur les bits ou l'état d'inhibition a changé (voir `ReceptionInfoNtfEvent`). Le message `ReceptionInfoNtf` est fourni à un client connecté en conséquence de l'abonnement aux notifications de changement d'état concernant les conditions de réception (messages `selectReceptionInfoReq` et `selectReceptionInfoCnf`).

Annexe A (informative)

Exemples de classes utilisées dans les API T-DMB

Des exemples de classes utilisées dans les API T-DMB sont décrits dans cette annexe.

```
SearchCnfEvent(  
T-DMBSource source,  
int result,  
int tuneState,  
int tuneFrequency,  
int transmissionMode,  
int synchronizationState)
```

```
SearchNtfEvent(  
T-DMBSource source,  
int tuneFrequency,  
int notifications)
```

```
ScanNtfEvent(  
T-DMBSource source,  
int tuneFrequency,  
int notifications)
```

```
ScanCnfEvent(  
T-DMBSource source,  
int result,  
int tuneState,  
int tuneFrequency,  
int transmissionModes,  
int noOfEnsemblesFound)
```

```
GetEnsembleInfoCnfEvent(  
T-DMBSource source,  
int result,  
EnsembleInfo ensembleInfo)
```

```
GetServiceInfoCnfEvent(  
T-DMBSource source,  
int result,  
ServiceInfo serviceInfo)  
GetComponentInfoCnfEvent(  
T-DMBSource source,  
int result,  
ComponentInfo componentInfo)
```

```
SelectReceptionInfoCnfEvent(  
T-DMBSource source,  
int result,  
boolean synchronizationNotifications,  
boolean bitErrorRateNotifications,  
boolean muteStateNotifications)
```

```
ReceptionInfoNtfEvent(  
T-DMBSource source,
```

```
int updateFlags,  
int synchronizationState,  
int bitErrorRateState,  
int muteState)
```

```
SelectComponentCnfEvent(  
T-DMBSource source,  
int result,  
ComponentId componentId,  
int selectionMode)
```

```
ObjectNtfEvent(  
T-DMBSource source,  
ComponentId componentId,  
ObjectId objectId,  
int selectionState,  
T-DMBObject object)
```

```
SelectObjectCnfEvent(  
T-DMBSource source,  
int result,  
ComponentId componentId,  
ObjectId objectId,  
int requestMode,  
boolean replaceSelections,  
Date accessTime)
```

```
ComponentInfo(  
ComponentId id,  
int type,  
byte[] data,  
boolean isPrimary,  
ServiceId[] parentIds,  
int accessControlSystem,  
boolean hasLabel,  
Label label,  
boolean hasLanguage,  
int language,  
boolean hasStartObjectId,  
ObjectId startObjectId,  
boolean hasObjectDirectoryId,  
ObjectId objectDirectoryId,  
boolean hasAudioComponent,  
ComponentId audioComponent,  
boolean hasBitrate,  
int bitrate)
```

```
ServiceInfo(  
ServiceId id,  
int type,  
EnsembleId parent,  
ComponentId[] componentIds,  
boolean isLocal,  
int accessControlSystem,  
boolean hasLabel,  
Label label,  
boolean hasLanguage,  
int language,  
boolean hasIsPrimary,
```

```
boolean isPrimary,  
boolean hasRegionId,  
int regionId,  
boolean hasRegionLabel,  
Label regionLabel,  
boolean hasStaticProgrammeType,  
ProgrammeType staticProgrammeType,  
boolean hasDynamicProgrammeType,  
ProgrammeType dynamicProgrammeType,  
boolean hasProgrammeNumber,  
ProgrammeNumber programmeNumber,  
boolean hasTimeOffset,  
int timeOffset,  
boolean hasAnnouncementSupport,  
AnnouncementSupport announcementSupport,  
boolean hasCountry,  
int country)
```

```
EnsembleInfo(  
EnsembleId id,  
ServiceId[] serviceIds,  
int frequency,  
int transmissionMode,  
boolean hasDate,  
Date date,  
boolean hasLabel,  
Label label,  
boolean hasCountry,  
int country)
```

Bibliographie

CEI 62104:2003, *Caractéristiques du récepteur DAB*

ISO 10486:1992, *Voitures particulières – Numéro d'identification des autoradios (CRIN)*

ETSI EN 50094:1992, *Access control system for the MAC/packet family: Eurocrypt* ETSI TS 101 993 v1.1.1 *Digital Audio Broadcasting (DAB); A Virtual Machine for DAB: DAB Java Specification*

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch