

MOST

Media Oriented Systems Transport

Multimedia and Control
Networking Technology

MOST FunctionBlock AuxIn

Rev 3.5

10/2008



Legal Notice

COPYRIGHT

© Copyright 1999 - 2008 MOST Cooperation. All rights reserved.

LICENSE DISCLAIMER

Nothing on any MOST Cooperation Web Site, or in any MOST Cooperation document, shall be construed as conferring any license under any of the MOST Cooperation or its members or any third party's intellectual property rights, whether by estoppel, implication, or otherwise.

CONTENT AND LIABILITY DISCLAIMER

MOST Cooperation or its members shall not be responsible for any errors or omissions contained at any MOST Cooperation Web Site, or in any MOST Cooperation document, and reserves the right to make changes without notice. Accordingly, all MOST Cooperation and third party information is provided "AS IS". In addition, MOST Cooperation or its members are not responsible for the content of any other Web Site linked to any MOST Cooperation Web Site. Links are provided as Internet navigation tools only.

MOST COOPERATION AND ITS MEMBERS DISCLAIM ALL WARRANTIES WITH REGARD TO THE INFORMATION (INCLUDING ANY SOFTWARE) PROVIDED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

In no event shall MOST Cooperation or its members be liable for any damages whatsoever, and in particular MOST Cooperation or its members shall not be liable for special, indirect, consequential, or incidental damages, or damages for lost profits, loss of revenue, or loss of use, arising out of or related to any MOST Cooperation Web Site, any MOST Cooperation document, or the information contained in it, whether such damages arise in contract, negligence, tort, under statute, in equity, at law or otherwise.

FEEDBACK INFORMATION

Any information provided to MOST Cooperation in connection with any MOST Cooperation Web Site, or any MOST Cooperation document, shall be provided by the submitter and received by MOST Cooperation on a non-confidential basis. MOST Cooperation shall be free to use such information on an unrestricted basis.

TRADEMARKS

MOST Cooperation and its members prohibit the unauthorized use of any of their trademarks. MOST Cooperation specifically prohibits the use of the MOST Cooperation LOGO unless the use is approved by the Steering Committee of MOST Cooperation.

SUPPORT AND FURTHER INFORMATION

For more information on the MOST technology, please contact:

MOST Cooperation

Administration
Bannwaldallee 48
D-76185 Karlsruhe
Germany

Tel: (+49) (0) 721 966 50 00

Fax: (+49) (0) 721 966 50 01

E-mail: contact@mostcooperation.com

Web: www.mostcooperation.com



© Copyright 1999 - 2008 MOST Cooperation.
All rights reserved.

MOST is a registered trademark

Contents

1	INTRODUCTION.....	10
2	FUNCTION CATALOG.....	10
2.1	AuxIn (FBlockID=0x24).....	10
2.1.1	VideoFrequency (0x131).....	14
2.1.2	VideoNorm (0x132).....	15
2.1.3	VideoFormat (0x135).....	16
2.1.4	DeckStatus (0x200).....	17
2.1.5	TimePosition (0x201).....	18
2.1.6	MediaPosition (0x204).....	20
2.1.7	TitlePosition (0x205).....	21
2.1.8	ChapterPosition (0x206).....	22
2.1.9	VideoInteraction (0x251).....	23
2.1.10	DeckEvent (0x430).....	24
2.1.11	MediaEvent (0x431).....	25
2.1.12	AuxDeviceInfo (0x432).....	27
2.1.13	DeviceBrowsingCapabilities (0x433).....	29
2.1.14	TrackInformation (0x434).....	31
2.1.15	AsyncControlSwitch (0x436).....	32
2.1.16	InputGainOffset (0x437).....	33
2.1.17	AuxTimeInformation (0x438).....	34
2.1.18	AuxTrackPosition (0x439).....	36
2.1.19	Random (0x450).....	37
2.1.20	Scan (0x451).....	38
2.1.21	Repeat (0x452).....	39
2.1.22	StringParameterMaxLength (0x490).....	41
2.1.23	SelectAudioListInfo (0x4A1).....	42
2.1.24	SelectAudioListFilter (0x4A2).....	53
2.1.25	CurrentAudioListFilter (0x4A3).....	58
2.1.26	SelectCurrentAudioListInfo (0x4A4).....	63
2.1.27	CurrentMediaListContentChanged (0x4A5).....	70
2.1.28	SelectCoverArt (0x4A6).....	71
2.1.29	SelectCurrentCoverArt (0x4A7).....	79
2.1.30	CoverArtSetup (0x4A8).....	84
2.1.31	RetrieveCoverArt (0x4A9).....	86
2.1.32	SelectAuxPlayer (0x4AA).....	88
2.1.33	AvailableAuxPlayer (0x4AB).....	89
2.1.34	AuxPlayerStatus (0x4AC).....	91
2.1.35	DeckStatusSet (0x4D0).....	92
2.1.36	MediaPositionSet (0x4D1).....	94
2.1.37	TitlePositionSet (0x4D2).....	95
2.1.38	ChapterPositionSet (0x4D3).....	96
2.1.39	TrackInformationSet (0x4D4).....	97
2.1.40	AuxTrackPositionSet (0x4D5).....	98
2.1.41	RandomSet (0x4D6).....	99
2.1.42	ScanSet (0x4D7).....	100
2.1.43	RepeatSet (0x4D8).....	101
3	DYNAMIC SPECIFICATION.....	102
3.1	AuxIn Startup.....	103
3.1.1	AuxIn_Initialization_Device_Connected.....	103
3.1.2	AuxIn_Initialization_Device_Not_Connected.....	104
3.1.3	AuxIn_Initialization_Async_Not_Supported.....	105
3.1.4	AuxIn_Standard_Notification_Set.....	106
3.1.5	AuxIn_Standard_Notification_Clear.....	108
3.2	Connecting an External Device.....	110

3.2.1	AuxIn_Hot_Plug	110
3.2.2	AuxIn_Hot_Unplug_NotActive	111
3.2.3	AuxIn_Hot_Unplug_Active	111
3.3	Audio Source Handling	112
3.3.1	AuxIn_Activation	112
3.3.2	AuxIn_Deactivation	112
3.4	Managing Media Lists	113
3.4.1	AuxIn_Browsing	113
3.4.2	AuxIn_Selection	114
3.5	AuxIn Control	115
3.5.1	AuxIn_Title_Selection	115
3.5.2	AuxIn_Set_Time_Position	116
3.5.3	AuxIn_DeckStatus_Play	116
3.5.4	AuxIn_DeckStatus_Stop	117
3.5.5	AuxIn_DeckStatus_Pause	118
3.5.6	AuxIn_DeckStatus_Search_Forward	119
3.5.7	AuxIn_DeckStatus_Search_Backward	120
3.5.8	AuxIn_Random	120
3.5.9	AuxIn_Scan	121
3.5.10	AuxIn_Repeat	122
3.5.11	AuxIn_Notify_Time_Position	122
3.5.12	AuxIn_Notify_Track_Change	123
3.5.13	AuxIn_CoverArt_Browsing	124
3.5.14	AuxIn_SelectMatchingCoverArt	125
3.5.15	AuxIn_RetrieveCoverArt	126
3.6	AuxPlayer	128
3.6.1	AuxIn_AuxPlayer_AvailablePlayerList	128
3.6.2	AuxIn_AuxPlayer_Selection	129

Document References

All documents which this MOST document have references to are listed here with the actual revision this document is referring to.

Number	Document	Revision
[1]	MOST Specification	2.5
[2]	GeneralFBlock Speed Grade MOST25	2.5.1
[3]	GeneralFBlock Speed Grade MOST50	2.5.1

Bibliography MOST Function Catalog

This is a list of released FBlocks at the release time of this specification. FBlocks which are released later are not reflected in this list.

FBlockID	FunctionBlock
-	GeneralFBlock
-	GeneralPlayer
0x01	NetBlock
0x02	NetworkMaster
0x03	ConnectionMaster
0x06	Diagnosis
0x0E	Tool
0x0F	Enhanced Testability
0x22	AudioAmplifier
0x24	AuxIn
0x26	MicrophoneInput
0x30	AudioTapePlayer
0x31	AudioDiskPlayer
0x34	DVDVideoPlayer
0x40	AmFmTuner
0x41	TMCTuner
0x42	TVTuner
0x43	DABTuner
0x44	SDARS
0x50	Telephone
0x51	GeneralPhoneBook
0x60	GraphicDisplay
-	Unique Functions

Document History

Changes AuxIn FBlock Rev 3.0-01 to AuxIn FBlock Rev 3.5

Change Ref.	FktID/ Section	Changes
3V5-001	General	Corrected some existing descriptions; adapted descriptions to changes listed in other Modifications. Corrected typos and made minor changes to increase readability and for clarification.
3V5-002	General	Moved description of SortOrder and FilterData to FBlock description as general concepts.
3V5-003	General	Bit 7 of MediaType and MediaTypeFilter renamed to "Video".
3V5-004	General	Fixed range values (e.g., one off in max. UnsignedLong).
3V5-005	General	"Set" OPTypes for functions 0x200, 0x204, 0x205, 0x206, 0x434, 0x439, 0x450, 0x451, 0x452 marked as deprecated.
3V5-006	0x000	Removed FktIDs and added reference to GeneralFBlock, where the function is contained already.
3V5-007	0x001	Removed Notification and added reference to GeneralFBlock, where the function is contained already.
3V5-008	0x002	Removed NotificationCheck and added reference to GeneralFBlock, where the function is contained already.
3V5-009	0x100	Removed SourceInfo and added reference to GeneralFBlock, where the function is contained already.
3V5-010	0x101	Removed Allocate and added reference to GeneralFBlock, where the function is contained already.
3V5-011	0x102	Removed DeAllocate and added reference to GeneralFBlock, where the function is contained already.
3V5-012	0x103	Removed SourceActivity and added reference to GeneralFBlock, where the function is contained already.
3V5-013	0x104	Removed SourceName and added reference to GeneralFBlock, where the function is contained already.
3V5-014	0x105	Removed SourceConnect and added reference to GeneralFBlock, where the function is contained already.
3V5-015	0x106	Removed SourceDisConnect and added reference to GeneralFBlock, where the function is contained already.
3V5-016	0x107	Removed SourceRouting and added reference to GeneralFBlock, where the function is contained already.
3V5-017	0x116	Removed SyncDataInfo and added reference to GeneralFBlock, where the function is contained already.
3V5-020	0x131	Added VideoFrequency function.
3V5-021	0x135	Added VideoFormat function.
3V5-022	0x132	Added VideoNorm function.
3V5-023	0x251	Added VideoInteraction function.
3V5-024	0x432	AuxDeviceInfo: Added 0x1F - Bluetooth to DeviceType.
3V5-025	0x433	DeviceBrowsingCapabilities: Added codes 11 through 15.
3V5-026	0x4A5	Added CurrentMediaListContentChanged function.
3V5-027	0x4A6	Added SelectCoverArt function.
3V5-028	0x4A7	Added SelectCurrentCoverArt function.
3V5-029	0x4A8	Added CoverArtSetup function.
3V5-030	0x4A9	Added RetrieveCoverArt function.
3V5-031	0x4A9	Added RetrieveCoverArt function.
3V5-032	0x4AA	Added SelectAuxPlayer function.
3V5-033	0x4AB	Added AvailableAuxPlayer function.

Change Ref.	FktID/ Section	Changes
3V5-034	0x4AC	Added AuxPlayerStatus function.
3V5-035	0x4D0	Added DeckStatusSet function.
3V5-036	0x4D1	Added MediaPositionSet function.
3V5-037	0x4D2	Added TitlePositionSet function.
3V5-038	0x4D3	Added ChapterPositionSet function.
3V5-039	0x4D4	Added TrackInformationSet function.
3V5-040	0x4D5	Added AuxTrackPositionSet function.
3V5-041	0x4D6	Added RandomSet function.
3V5-042	0x4D7	Added ScanSet function.
3V5-043	0x4D8	Added RepeatSet function.
3V5-044	3	Dynamic Specification chapter completely reworked.

Changes AuxIn FBlock Rev 3.0 to AuxIn FBlock Rev 3.0-01

Change Ref.	Section	Changes
3V0-01-001	All	Minor text formatting and layout changes.
3V0-01-002	3	Added missing MSC: AUXIn01.02_AUXIn_Active.
3V0-01-003	3	Added missing MSC comments.

Changes AuxIn FBlock Rev 2.4 to AuxIn FBlock Rev 3.0

Change Ref.	Section	Changes
3V0-001	All	Totally revised.

Changes AuxIn FBlock Rev 2.3 to AuxIn FBlock Rev 2.4

Change Ref.	Section	Changes
2V4-001	2.1.20	Extension of Enum for DeckStatus.Play to use the full enum range from GeneralPlayer
2V4-002	2.1.31	Added Random, Repeat and Scan from GeneralPlayer
2V4-003	2.1.32	New function MediaLibraryName to uniquely identify more than one instance of AuxIn FBlock
2V4-004	2.1.33	New function AudioListInfo as a generic database retrieval function for audio specific data
2V4-005	2.1.34	New function AudioListFilter to set the player to play a specific selection out of the full list of audio media titles
2V4-006	2.1.35	New function MediaPosition as an extension of TrackPosition to allow access to a large number of media titles (>65000)
2V4-007	2.1.36	New function AsyncControlSwitch to optionally switch the message transport channel to use the asynchronous channel. This is useful for large database volumes. Extended AuxInProfile to reflect Database Retrieval
2V4-008	2.1.37	New function InputGainOffset to allow the device to exert volume control of an analog input line
2V4-009	2.1.38	New function CurrentAudioListInfo to retrieve information about the currently playing audio titles
2V4-010	2.1.39	AuxProfile changed to Profile Extended Profile to reflect Database Retrieval Profile is now of Section type "Extension"

1 Introduction

This document contains the specification of an FBlock. MOST FBlocks are standardized and maintained by MOST workgroup Device Architecture (WG_DA). In order to speed up the process of making new FBlocks available, every FBlock will be updated individually as required.

2 Function Catalog

2.1 AuxIn (FBlockID=0x24)

Via the AuxIn Function Block a controller in the OEM MOST is able to control consumer electronic devices. AuxIn is derived from the GeneralPlayer (V2.4.2) and provides a subset of the functions defined there and additional functions to enable sophisticated playback and media browsing experience. This FBlock is based on MOST specification V2.5. It obeys the DTD 6.05. The general MOST specific error messages apply. The following codes shall be used in general:

- ErrorCode 0x20, ErrorInfo 0x04 "Value not available": If the property Set OpType was not successful, e.g. due to TrackPosition.Set value out of range, whereas parameter errors will be signaled by ErrorCode 0x07.
- ErrorCode 0x40 "Busy": If the called method could not be handled at the current time and the controller shall try later with the same parameters.
- ErrorCode 0x41 "Not available": If the called method is implemented but not available at the moment.
- ErrorCode 0x07 "Parameter not available" (see next line):
- ErrorCode 0x43 "Method aborted": If method is aborted with an "Abort" function call. If the called method could not be handled and the controller shall not ask again with the same parameters without receiving an Abort, ErrorCode 0x07 "Parameter not available" shall be signaled.

Description of common concepts

SortOrder

SortOrder1, SortOrder2 and SortOrder3 indicate which record element of AudioMediaData is to be used for ordering result. The parameters are applied subsequently 1 to 3 with the rule that a later SortOrder only sorts entries that are equal after the previous application of SortOrder criteria. E.g. first order after genres and in titles with equal genres sort for artists. Positive numbers of SortOrder[1,2,3] indicate ascending order, negative numbers indicate descending order. See ResultData for the mapping between number and property. The number used is equal to the bit number of ResultData (counting bits subsequently through ResultsData1, ResultsData2, ResultsData3). Example: A value of 15 will sort the result by Filename in ascending order. A value of 0 will apply default sort order of the device. SortOrder[1,2,3] values from -127..-100 and 100..127 are reserved for OEM specific ordering scheme.

FilterData

FilterData is used for setting a filter for record elements to match certain criteria. The request field is always matched against the contents of the field. Only records matching in the fields provided are considered further. Wildcard matching is supported. There are three possible matching systems to be applied: Numbers, Strings and BoolFields.

Numbers:

- empty will match anything (any number or an empty field)
- * will match any number
- 2 will match 2
- 1< will match any number larger than 1
- <2 will match 1 and 0
- 1<4 will match numbers 2 and 3
- !<expr> will negate the result of expression, e.g. !2 will match anything but 2.

Strings:

- empty will match anything (any string or an empty string)
- * will match any string
- a will match exactly A (not case sensitive)
- *a* will match anything that contains an a
- alpha* will match anything that starts with alpha
- *a will match anything that ends with a
- *alpha will match anything that ends with alpha
- :n will match the nth element in the previous result array. n is determined by a text-to-number conversion as produced by scanf(":%d"). The first element is numbered ":1". The order in which filters are applied is implementation dependent.
- !<expr> will negate the result of expression, e.g. !A* will match anything that does not start with A

Function Overview		
FktID	Name	Section Type
0x131	VideoFrequency	Coordination
0x132	VideoNorm	Coordination
0x135	VideoFormat	Coordination
0x200	DeckStatus	Mandatory
0x201	TimePosition	Mandatory
0x204	MediaPosition	Mandatory
0x205	TitlePosition	Mandatory
0x206	ChapterPosition	Mandatory
0x251	VideoInteraction	Mandatory
0x430	DeckEvent	Extension
0x431	MediaEvent	Extension
0x432	AuxDeviceInfo	Extension
0x433	DeviceBrowsingCapabilities	Extension
0x434	TrackInformation	Extension
0x436	AsyncControlSwitch	Extension
0x437	InputGainOffset	Extension
0x438	AuxTimeInformation	Extension
0x439	AuxTrackPosition	Extension
0x450	Random	Extension
0x451	Scan	Extension
0x452	Repeat	Extension
0x490	StringParameterMaxLength	Extension
0x4A1	SelectAudioListInfo	Extension
0x4A2	SelectAudioListFilter	Extension
0x4A3	CurrentAudioListFilter	Extension
0x4A4	SelectCurrentAudioListInfo	Extension
0x4A5	CurrentMediaListContentChanged	Extension
0x4A6	SelectCoverArt	Extension
0x4A7	SelectCurrentCoverArt	Extension
0x4A8	CoverArtSetup	Extension
0x4A9	RetrieveCoverArt	Extension
0x4AA	SelectAuxPlayer	Extension
0x4AB	AvailableAuxPlayer	Extension
0x4AC	AuxPlayerStatus	Extension
0x4D0	DeckStatusSet	Extension
0x4D1	MediaPositionSet	Extension
0x4D2	TitlePositionSet	Extension
0x4D3	ChapterPositionSet	Extension
0x4D4	TrackInformationSet	Extension
0x4D5	AuxTrackPositionSet	Extension
0x4D6	RandomSet	Extension
0x4D7	ScanSet	Extension
0x4D8	RepeatSet	Extension

2.1.1 VideoFrequency (0x131)

Section Type: Coordination

The frequency of the video signal. This function is available and set only if analog video output is available.

2.1.1.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AuxIn (0x24)	VideoFrequency (0x131)	Get	-
		Status	VideoFrequency
		Error	ErrorCode, ErrorInfo

2.1.1.2 Parameter

VideoFrequency

Basis datatype	Range of values	Code	Description
Enum	0x00..0x02	0x00	unknown
		0x01	50 Hz
		0x02	60 Hz

2.1.2 VideoNorm (0x132)

Section Type: Coordination

The norm used for this video signal. This function is available and set only if analog video output is available.

2.1.2.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AuxIn (0x24)	VideoNorm (0x132)	Get	-
		Status	VideoNorm
		Error	ErrorCode, ErrorInfo

2.1.2.2 Parameter

VideoNorm

Enumeration of the available television systems.

Basis datatype	Range of values	Code	Description
Enum	0x00..0x2A	0x00	unknown
		0x01	EIA
		0x02	CCIR
		0x09	NTSC
		0x0A	NTSC-EUROPE
		0x0B	NTSC-M
		0x0C	NTSC-JAPAN
		0x13	PAL
		0x14	PAL-BG
		0x15	PAL-I
		0x16	PAL-M
		0x17	PAL-N
		0x18	PAL-DK
		0x19	PAL-AUSTRALIA
		0x1A	PAL-ITALIA
		0x1B	PAL-MAROCCO
		0x1C	PAL-VRC
		0x1E	SECAM-BG
		0x1F	SECAM-DK
		0x20	SECAM-K1
		0x21	SECAM-L
0x28	HDTV		
0x29	MUSE(JAPAN)		
0x2A	HDTV(USA)		

2.1.3 VideoFormat (0x135)

Section Type: Coordination

The format of the video signal. This function is available and set only if analog video output is available.

2.1.3.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AuxIn (0x24)	VideoFormat (0x135)	Set	VideoFormat
		Get	-
		SetGet	VideoFormat
		Status	VideoFormat
		Error	ErrorCode, ErrorInfo

2.1.3.2 Parameter

VideoFormat

Basis datatype	Range of values	Code	Description
Enum	0x00..0x04	0x00	unknown
		0x01	standard
		0x02	wide
		0x03	Letter box
		0x04	Pan SCAN

2.1.4 DeckStatus (0x200)

Section Type: Mandatory

This property controls and shows the state of the drive. For connectivity devices, this property shows the state of the connected device, it assumes that a device is connected and therefore does not make any statement about the connectivity status. If no device is connected, the DeckStatus is undefined (i.e. it can be anything, even an error). The default value is 'Load / Connect'.

Note: The use of OPType Set for this function is not recommended. Use method DeckStatusSet (0x4D0) to set the value.

2.1.4.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AuxIn (0x24)	DeckStatus (0x200)	Set	DeckStatus
		Get	-
		SetGet	DeckStatus
		Status	DeckStatus
		Error	ErrorCode, ErrorInfo

2.1.4.2 Parameter

DeckStatus

- 0x00..0x1F = general states
- 0x20..0x2F = video specific states
- 0x30..0x3F = tape specific states
- 0x40..0x4F = file handling
- 0x50..0x5F = recording

Basis datatype	Range of values	Code	Description
Enum	0x00..0x50	0x00	Play
		0x01	Stop
		0x02	Pause
		0x03	Load / Connect
		0x04	Unload / Disconnect
		0x05	Search Forward
		0x06	Search Backward
		0x07	Fast Forward
		0x08	Fast Backward
		0x09	Empty / No titles available
		0x0A	Retract
		0x20	Slow Motion Forward
		0x21	Slow Motion Backward
		0x22	StepbyStep
		0x23	PreStop
		0x30	Rewind to Begin of Tape
		0x31	Forward to End of Tape
		0x32	Search Startposition next title
		0x33	Search Startposition last Title
		0x40	Fileplay
0x41	Filetransfer		
0x50	Record		

2.1.5 TimePosition (0x201)

Section Type: Mandatory

Display current position or set a new position in milliseconds. Datatype: Record of {DiskTime,TrackTime, TitleTime} Remark: * an event is triggered only every second.

2.1.5.1 Format of Function

Function classes: Record of { Number Number Number }

FBlock	Function	OPType	Parameter
AuxIn (0x24)	TimePosition (0x201)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Increment	Pos, NSteps
		Decrement	Pos, NSteps
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.5.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Since this is an unidimensional construction, the second Byte y is unused (y=0=const) and the simplified notation Pos={x} is valid. Valid range: x=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
		Pos	Data
Record	-	{ x=0 }	DiskTime, TrackTime, TitleTime
		{ x=1 }	DiskTime
		{ x=2 }	TrackTime
		{ x=3 }	TitleTime

DiskTime

Time details in milliseconds related to begin of disk.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Long	0		1	ms

TrackTime

Time details in milliseconds related to begin of current track.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Long	0		1	ms

TitleTime

Time details in milliseconds related to the start of the current title

Basis datatype	Exp.	Range of values	Step	Unit
Signed Long	0		1	ms

NSteps

Number of steps for adjustment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	1..255	1	none

2.1.6 MediaPosition (0x204)

Section Type: Mandatory

Displays the current or sets a new position as audio medium. MediaPosition differs from TrackPosition in the way that MediaPosition refers to the absolute title by using a unique identifier while TrackPosition refers by index to the relative title within the list of currently playing titles.

Remark: Since Tag is a unique identifier used by the media library, it does not make sense to increase or decrease this tag.

When MediaPosition is set to a value which is not a member of the current filter, an error is returned. Notification must be possible.

Note: The use of OPType Set for this function is not recommended. Use method MediaPositionSet (0x4D2) to set the value.

2.1.6.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
AuxIn (0x24)	MediaPosition (0x204)	Set	Tag
		Get	-
		SetGet	Tag
		Status	Tag
		Error	ErrorCode, ErrorInfo

2.1.6.2 Parameter

Tag

Tag is a unique identifier within the media library.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

2.1.7 TitlePosition (0x205)

Section Type: Mandatory

Display current position or set a new position. When TitlePosition is set to a value which is not a member of the current filter, an error is returned. Notification must be possible.

Note: The use of OPType Set for this function is not recommended. Use method TitlePositionSet (0x4D3) to set the value.

2.1.7.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
AuxIn (0x24)	TitlePosition (0x205)	Set	TitlePosition
		Get	-
		SetGet	TitlePosition
		Increment	NSteps
		Decrement	NSteps
		Status	TitlePosition
		Error	ErrorCode, ErrorInfo

2.1.7.2 Parameter

TitlePosition

Number of steps for adjustment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

NSteps

Number of steps for adjustment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	1..255	1	none

2.1.8 ChapterPosition (0x206)

Section Type: Mandatory

Displays the current position or sets a new position as chapter, e.g. on a DVD or Audiobook. When ChapterPosition is set to a value which is not a member of the current filter, an error is returned. Notification must be possible.

Note: The use of OPType Set for this function is not recommended. Use method ChapterPositionSet (0x4D4) to set the value.

2.1.8.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
AuxIn (0x24)	ChapterPosition (0x206)	Set	ChapterPosition
		Get	-
		SetGet	ChapterPosition
		Increment	NSteps
		Decrement	NSteps
		Status	ChapterPosition
		Error	ErrorCode, ErrorInfo

2.1.8.2 Parameter

ChapterPosition

Number of steps for adjustment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

NSteps

Number of steps for adjustment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	1..255	1	none

2.1.9 VideoInteraction (0x251)

Section Type: Mandatory

Operates the DVD menu

2.1.9.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	VideoInteraction (0x251)	Start	VideoInteraction
		StartResult	VideoInteraction
		Processing	-
		Result	-
		Error	ErrorCode, ErrorInfo

2.1.9.2 Parameter

VideoInteraction

Basis datatype	Range of values	Code	Description
Enum	0x00..0x09	0x00	Title
		0x01	Root
		0x02	Enter
		0x03	Up
		0x04	Down
		0x05	Right
		0x06	Left
		0x07	Back
		0x08	Previous Chapter
		0x09	Next Chapter

2.1.10 DeckEvent (0x430)

Section Type: Extension

Events concerning the drive or device. Whether the connected device/drive can signal the events to the gateway depends on the characteristics of the external device.

Value	Description
OverTemperature / UnderTemperature	external device signals temperature problems.
OverVoltage / UnderVoltage	external device signals power problems.
TooMuchPowerDrawn	external device draws too much power from AuxIn gateway.

2.1.10.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AuxIn (0x24)	DeckEvent (0x430)	Get	-
		Status	DeckEvent
		Error	ErrorCode, ErrorInfo

2.1.10.2 Parameter

DeckEvent

Type of event

Basis datatype	Range of values	Code	Description
Enum	0x00..0x06	0x00	Normal Operation
		0x01	Deck Error
		0x02	Over normal Temperature
		0x03	Under Normal Temperature
		0x04	Under Voltage
		0x05	Over Voltage
		0x06	TooMuchPowerDrawn

2.1.11 MediaEvent (0x431)

Section Type: Extension

Events concerning the current medium that occur during playback of tracks. The usage and generation of events is deployment specific. As rule of thumb the following recommendation are given: The implementation should avoid very fast toggling between different values and should therefore implement a minimum time of status hold time (e.g. 200 ms).

- NormalOperation: set as default if none of the other events are present
- Corrupted File or Track: signaled if the current track/title could not be played due to corruption (e.g. bit errors or unknown DRM format); implementation specific recovery strategy may apply (e.g. jump to next track/title or stop playback)
- File or Track Change Delayed: should be signaled if track change is delayed (time is implementation specific)
- EndOfTracklist: Signaled if tracklist has reached the end and DeckStatus changes to "Pause" or "Stop"
- SecuredFile: signaled if file is secured (e.g. DRM protection recognized) and file could not be played
- File/Track not found: signaled if a track inside a playlist (e.g. m3u file) is not found; implementation specific behavior of recovery strategy
- Filter with no Media File/Track: signaled if the set filter (SelectAudioListFilter) covers no media files (e.g. empty directory or directory with no media files)

2.1.11.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AuxIn (0x24)	MediaEvent (0x431)	Get	-
		Status	MediaEvent
		Error	ErrorCode, ErrorInfo

2.1.11.2 Parameter

MediaEvent

Type of event

Basis datatype	Range of values	Code	Description
Enum	0x00..0x12	0x00	Normal Operation
		0x01	Corrupted File or Track
		0x02	End of CD
		0x03	Corrupted ROM Filesystem
		0x04	Startposition
		0x05	TOC Unreadable
		0x06	Disk not available
		0x07	End of File
		0x08	File or Track Change Delayed
		0x09	End of Tracklist
		0x10	Secured file (e.g. DRM)
		0x11	File/Track not found
		0x12	Filter with no MediaFile/Track

2.1.12 AuxDeviceInfo (0x432)

Section Type: Extension

This is a collection of properties that gives information on a device connected to AuxIn device.

Notification possible

2.1.12.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AuxIn (0x24)	AuxDeviceInfo (0x432)	Get	-
		Status	DeviceNum , DeviceClass , DeviceType , DeviceName
		Error	ErrorCode, ErrorInfo

2.1.12.2 Parameter

DeviceNum

Identifies the gateway device.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

DeviceClass

Type of the device.

None (0x00) -> if no device is connected

Unknown (0x01) -> if something is connected but not correctly recognized or initialized

Database Device (0x02) -> browsing is done primarily through metatags other than directory/filename (e.g. Title, Artist)

MassStorageDevice (0x03) -> browsing is done exclusively through directory/file navigation

Aux Player (0x04) -> Line-In

Error (0x05) -> None of the values above

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	None
		0x01	Unknown
		0x02	Database Device
		0x03	Mass Storage Device
		0x04	Aux Player
		0x05	Error

DeviceType

Type of the device.

0x00..0x20 -> if initialization is finished and access to the device is possible (DeckStatus indicates the playback status!)

0x30 -> not supported device; e.g. USB WLAN/BT stick

0x31 -> Device not readable, e.g. file system corrupted

0x32 -> if too many device are connected; e.g. two devices at the same time

0x33 -> any other error

Basis datatype	Range of values	Code	Description
Enum	0x00..0xff	0x00	iPod Serial
		0x01	iPod USB
		0x02	MTP Analog
		0x03	MTP Digital
		0x04..0x0d	Reserved
		0x0e	Virtual Server
		0x0f	HDD
		0x10	Mass Storage Device
		0x11	DVD
		0x12	CD
		0x13..0x1e	Reserved
		0x1f	Bluetooth
		0x20	AuxIn/LinIn
		0x21..0x2f	Reserved
		0x30	Not Supported
		0x31	Unreadable
		0x32	TooManyDevices
		0x33	Unspecified Error
0x34..0xfe	Reserved		
0xff	Not set		

DeviceName

Name of the connected device.

Basis datatype	MaxSize
String	30

2.1.13 DeviceBrowsingCapabilities (0x433)

Section Type: Extension

Provides information about the browsing capabilities of the connected devices for display at the HMI. The return value is a mixture of different bits set a boolean values to allow dynamic changes in the HMI for displaying browsing capabilities. The browsing capabilities could be also updated during runtime, e.g. after scanning a Mass Storage Device playlists are detected and browsing using playlists is made possible. Other capabilities are depending on the firmware of certain connected devices, e.g Audiobooks on iPods.

Notification possible

2.1.13.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AuxIn (0x24)	DeviceBrowsing Capabilities (0x433)	Get	-
		Status	DeviceBrowsingCapabilities
		Error	ErrorCode, ErrorInfo

2.1.13.2 Parameter

DeviceBrowsingCapabilities

Basis datatype	Bit #	Code	Description
Unsigned Word	Bit 0	False	not available
		True	Title
	Bit 1	False	not available
		True	Artist
	Bit 2	False	not available
		True	Album
	Bit 3	False	not available
		True	Year
	Bit 4	False	not available
		True	Genre
	Bit 5	False	not available
		True	Composer
	Bit 6	False	not available
		True	Filename
	Bit 7	False	not available
		True	Playlist
	Bit 8	False	not available
		True	Audiobook
	Bit 9	False	not available
		True	Podcast
	Bit 10	False	not available
		True	Compilations
	Bit 11	False	not available
		True	Opus
	Bit 12	False	not available
		True	Soloist
	Bit 13	False	not available
		True	Conductor
Bit 14	False	not available	

Basis datatype	Bit #	Code	Description
		True	Ensemble
	Bit 15	False	not available
		True	SimpleVideo

2.1.14 TrackInformation (0x434)

Section Type: Extension

Provides information about the number of tracks in the current played directory or category without subdirectories. It enables the display of information like Song x of y in a directory or category. CurrentNumberTracks provides total number of tracks (excluding playlists) in the currently played directory or category. This property cannot be set as it is defined by the current directory or category structure. CurrentRelativeTrackPosition provides the relative trackposition in the current directory or category.

Increment and decrement should implement a wrap-around behaviour, i.e. increment when CurrentNumberTracks = CurrentRelativeTrackPosition leads to CurrentRelativeTrackPosition = 1 (and vice versa for decrement).

In case of database players the mapping between TrackInformation and TotalNumberResults/TrackPosition is as follows:

CurrentNumberTracks = TotalNumberResults (as it is returned by CurrentAudioListInfo)

CurrentRelativeTrackPosition = TrackPosition

Notification possible.

Note: The use of OPType Set for this function is not recommended. Use method TrackInformationSet (0x4D5) to set the value.

2.1.14.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AuxIn (0x24)	TrackInformation (0x434)	Set	CurrentRelativeTrackPosition
		Get	-
		SetGet	CurrentRelativeTrackPosition
		Increment	NStepsLong
		Decrement	NStepsLong
		Status	CurrentNumberTracks , CurrentRelativeTrackPosition
		Error	ErrorCode, ErrorInfo

2.1.14.2 Parameter

CurrentRelativeTrackPosition

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

NStepsLong

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

CurrentNumberTracks

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

2.1.15 AsyncControlSwitch (0x436)

Section Type: Extension

Due to the high amount of data that can be transferred when calling certain functions of AuxIn FBlock this switch determines whether some or all OPTypes of certain functions use the Packet Data Channel instead of the Control Channel. See the function definitions to determine which functions and OPTypes are affected by this switch.

The hardware is not required to support the Packet Data Channel, in this case, AsyncControlSwitch will respond with an error. The Default on every startup, Configuration.Status(NotOK), etc. must be "False". Before using the Packet Data Channel, the controller must Set or SetGet this property to "True"; and must receive a Status of "True" for confirmation. The controller may Set or SetGet this property to "True" only if it supports the Packet Data Channel itself. If several controllers access the same AuxIn FBlock, it is advised to set a Notification for this property and have each controller send AsyncControlSwitch.Set either with "True" or "False" depending on the abilities of the controller. This will ensure that switching to Packet Data Channel is only done when all controllers (and the FBlock) support packet data handling.

Notification must be possible.

2.1.15.1 Format of Function

Function classes: Switch

FBlock	Function	OPType	Parameter
AuxIn (0x24)	AsyncControl Switch (0x436)	Set	Async
		Get	-
		SetGet	Async
		Status	Async
		Error	ErrorCode, ErrorInfo

2.1.15.2 Parameter

Async

Determines whether requests and responses (status and error messages) for AudioListInfo and CurrentAudioListInfo are sent through the Packet Data Channel instead of the Control Channel.

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	True	Packet Data Channel
		False	Control Channel

2.1.16 InputGainOffset (0x437)

Section Type: Extension

Sets the audio level gain for the analog line-in of the AuxIn gateway in order to adapt to different analog input levels of different devices. This is similar to InputGainOffset (0x450) as used in the AudioAmplifier FBlock (0x22), but with only one (external/analog) input line.

2.1.16.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
AuxIn (0x24)	InputGainOffset (0x437)	Set	GainOffset
		Get	-
		SetGet	GainOffset
		Status	GainOffset
		Error	ErrorCode, ErrorInfo

2.1.16.2 Parameter

GainOffset

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	dB

2.1.17 AuxTimeInformation (0x438)

Section Type: Extension

Display of total time information of the current chapter, track or title in milliseconds.

Datatype: Record of {DiskTime,TrackTime, TitleTime}

Remark: The total playtime might change during playback due to estimation in case of variable bitrate encoding.

2.1.17.1 Format of Function

Function classes: Record of { Number Number Number }

FBlock	Function	OPType	Parameter
AuxIn (0x24)	AuxTime Information (0x438)	Get	Pos
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.17.2 Parameter

Pos

The parameter Pos={x,y} consists of two byte x and y and shows which parameter shall be set, inquired or read. Since this is an unidimensional construction, the second Byte y is unused (y=0=const) and the simplified notation Pos={x} is valid. Valid range: x=0..3

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

Data

The content of Data depends on parameter Pos={x,y}.

Basis datatype	Length	Description	
Record	-	Pos	Data
		{ x=0 }	TotalChapterTime, TotalTrackTime, TotalTitleTime
		{ x=1 }	TotalChapterTime
		{ x=2 }	TotalTrackTime
		{ x=3 }	TotalTitleTime

TotalChapterTime

Total playtime of the current chapter in milliseconds.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	ms

TotalTrackTime

Total playtime of the current track in milliseconds.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	ms

TotalTitleTime

Total playtime in milliseconds of the current title

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	ms

2.1.18 AuxTrackPosition (0x439)

Section Type: Extension

Displays the current position or sets a new position as track.

Note:

- Track=0 for "no track", e.g. if there is no medium available.
- For a tape player, Track=1 corresponds to the first side of the tape and Track=2 corresponds to the second side.
- The currently valid range of values is dependent on the medium. When required, the actual range is queriable by the interface.
- 0xFFFF FFFF: exact track number not known (e.g., AuxIn device is still calculating track number)

The use of OPType Set for this function is not recommended. Use method AuxTrackPositionSet (0x4D6) to set the value.

2.1.18.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
AuxIn (0x24)	AuxTrackPosition (0x439)	Set	Track
		Get	-
		SetGet	Track
		Increment	NStepsLong
		Decrement	NStepsLong
		Status	Track
		Error	ErrorCode, ErrorInfo

2.1.18.2 Parameter

Track

The current track.

special values see above

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	none

NStepsLong

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

2.1.19 Random (0x450)

Section Type: Extension

This property is for switching on and off the Random function. Some values are marked as optional.

Value	Description
Disk	Random selection within the current medium, e.g., disk.
Magazine	Random selection within the current magazine.
All / All Magazine	Random selection within all tracks on the medium or magazines.
Folder	Random selection within the current folder/directory (optional).
SubFolder	Random selection within the current folder/directory including subfolder (optional).
Tracklist	Random selection within the current selected tracklist.

Note: The use of OPType Set for this function is not recommended. Use method RandomSet (0x4D7) to set the value.

2.1.19.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AuxIn (0x24)	Random (0x450)	Set	RandomState
		Get	-
		SetGet	RandomState
		Status	RandomState
		Error	ErrorCode, ErrorInfo

2.1.19.2 Parameter

RandomState

Setting the RandomState to "All" can cause a implicit change of the used filter for playback, if the filter was not set to all media on the connected device. The changed filter can be signaled by the gateway using notification for CurrentAudioListFilter property.

Basis datatype	Range of values	Code	Description
Enum	0x00..0x0C	0x00	Off
		0x01	reserved
		0x02	Disk
		0x03	Magazine / Albums
		0x04	All / All Magazines
		0x05..0x09	Reserved
		0x0A	Folder
		0x0B	SubFolder
		0x0C	Tracklist

2.1.20 Scan (0x451)

Section Type: Extension

This property is for switching on and off the Scan function Some values are marked as optional.

Value	Description
Disk	Random selection within the current medium, e.g., disk.
Magazine	Random selection within the current magazine.
All / All Magazine	Random selection within all tracks on the medium or magazines.
Folder	Random selection within the current folder/directory (optional).
SubFolder	Random selection within the current folder/directory including subfolder (optional).
Tracklist	Random selection within the current selected tracklist.

Note: The use of OPType Set for this function is not recommended. Use method ScanSet (0x4D8) to set the value.

2.1.20.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AuxIn (0x24)	Scan (0x451)	Set	ScanState
		Get	-
		SetGet	ScanState
		Status	ScanState
		Error	ErrorCode, ErrorInfo

2.1.20.2 Parameter

ScanState

Basis datatype	Range of values	Code	Description
Enum	0x00..0x0C	0x00	Off
		0x01	reserved
		0x02	Disk
		0x03	Magazine
		0x04	All / All Magazines
		0x05..0x09	Reserved
		0x0A	Folder
		0x0B	SubFolder
		0x0C	Tracklist

2.1.21 Repeat (0x452)

Section Type: Extension

This property is for switching on and off the repeat function.

Value	Description
Track	Automatic repeat of the current track.
Disk	Automatic repeat of the current medium.
Magazine	Automatic repeat of all media of the current magazine.
All / All Magazines	Automatic repeat of all tracks on the medium or magazine.
Chapter	Automatic repeat of the current chapter.
Title	Automatic repeat of the current title.
AB	Automatic repeat of the marked section.
Folder	Automatic repeat within the current folder/directory (optional).
SubFolder	Automatic repeat within the current folder/directory including subfolder (optional).
Tracklist	Automatic repeat of the currents selected tracklist.

Note: The use of OPType Set for this function is not recommended. Use method RepeatSet (0x4D9) to set the value.

2.1.21.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
AuxIn (0x24)	Repeat (0x452)	Set	RepeatState
		Get	-
		SetGet	RepeatState
		Status	RepeatState
		Error	ErrorCode, ErrorInfo

2.1.21.2 Parameter

RepeatState

Setting the RepeatState to "All" can cause a implicit change of the used filter for playback, if the filter was not set to all media on the connected device. The changed filter can be signaled by the gateway using notification for CurrentAudioListFilter property.

Basis datatype	Range of values	Code	Description
Enum	0x00..0x0C	0x00	Off
		0x01	Track
		0x02	Disk
		0x03	Magazine
		0x04	All / All magazines
		0x05	Chapter
		0x06	Title
		0x07	AB
		0x08..0x09	Reserved
		0x0A	Folder

Basis datatype	Range of values	Code	Description
		0x0B	SubFolder
		0x0C	Tracklist

2.1.22 StringParameterMaxLength (0x490)

Section Type: Extension

With this property the controller sets the maximal length of string parameters (in characters) for audio meta data and filter data for compressed audio in order to reduce bus load. It applies in the functions 0x4A1 to 0x4A8 and 0x4B1 to 0x4B5.

This property does not explicitly apply to the DirectoryName (AudioMediaData) and DirectoryNameFilter (FilterData) parameter since the directory name is a concatenated string which must not to be shortened.

The slave performs duplicate detection of entries before cutting string length.

Warning: By applying the string length reduction the unambiguousness of entries might be lost, e.g. different entries might have the same string entry. They can still be distinguished by using the index notation (e.g: "[ESC]2")

The default value is 0xFFFF.

2.1.22.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
AuxIn (0x24)	StringParameter MaxLength (0x490)	Set	MaxLength
		SetGet	MaxLength
		Status	MaxLength
		Error	ErrorCode, ErrorInfo

2.1.22.2 Parameter

MaxLength

The maximum string length in characters that are sent by the slave to the controller in the affected functions. The string length on the bus depends of the string coding scheme.

The value 0xFFFF (default) means maximum string length according to MOST specification, i.e. no cutting of strings in the gateway.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

2.1.23 SelectAudioListInfo (0x4A1)

Section Type: Extension

Get audio specific information from a (possibly virtual) audio media information database on the AuxIn FBlock or on the device represented by the AuxIn FBlock. The returned list contains a thesaurus of audio specific information of all audio media available. This function allows complete querying of the media library with just one command.

Notes:

- The record used for the AudioMediaData parameter can be considered as a (non-optimized, redundant) table in a relational database
- If the filter is set and produces duplicate entries, the duplicates must be removed before transfer
- Record elements which are not requested should be empty (string: empty string, numbers: zero)
- It is advised to set the ResultData bit for PlayList with caution. Requesting all PlayLists with all Songs can return a huge number of result records: If a Song is in more than one PlayList, it will be returned once for each PlayList it is in (RIGHT OUTER JOIN operation between playlists and songs).
- If NumberResults is a large value and the FBlock does not have enough memory available to process the request, use ErrorCode 0x20, ErrorInfo 0x02 (List overflow) to indicate that the FBlock is not able to handle this request.
- If an optional filter method is used by the controller but the FBlock does not support this type of filter, use ErrorCode 0x07 (Parameter not available).

Application hints:

Mapping directories/files to a generic directory/playlist/filename mechanism

DirectoryName identifies the directory path, filename accordingly the name of the object (directory, audio file or playlist) which will be further identified using MediaType.

Usage examples:

```
AudioListInfo.StartResultAck(...,ResultData={filename,
directoryname},FilterData={...,DirectoryNameFilter="/topleveldir /firstleveldir/secondleveldir/",
MediaType = b11111111...}, )
will return all objects in a given directory (subdirectories, playlists, audio files).
```

```
AudioListInfo.StartResultAck(...,ResultData={filename, directoryname},
FilterData={...,DirectoryNameFilter="/topleveldir /firstleveldir/secondleveldir/*",
MediaTypeFilter=b00000010}, ...)
will return all directories starting with /topleveldir/firstleveldir/secondleveldir/, ie. the directory name
itself and all subdirectories.
```

Example answer:

```
/topleveldir/firstleveldir/secondleveldir/thirdleveldir1/
/topleveldir/firstleveldir/secondleveldir/thirdleveldir2/
/topleveldir/firstleveldir/secondleveldir/thirdleveldir3/
```

Playlists in a directory:

```
AudioListInfo.StartResultAck(...,ResultData={filename, directoryname},
FilterData={...,DirectoryNameFilter="/topleveldir /firstleveldir/secondleveldir/",
MediaTypeFilter=b00001000}, ...)
```

Files in a Playlist:

```
AudioListInfo.StartResultAck(...,ResultData={filename, directoryname, playlistname},
FilterData={..PlaylistNameFilter= "GivenPlaylistName" ,DirectoryNameFilter="/topleveldir
```

/firstleveldir/secondleveldir/", MediaTypeFilter=b01000000}, ...)

The FBlock allows a maximum directory pathname length of 255 characters. The controller must be aware of directory browsing method since filling of the appropriate entries like DirectoryNameFilter must be accomplished. Long playlist names must be supported adequately. A directory history can be implemented in the controller (including upDirectory or previousDirectory) by parsing the directoryname.

All implementations: Playlists found in the directory structure (e.g. m3u) should be presented as additional playlists. If they are presented without directory structure then property BrowsingCapabilities should indicate playlist browsing. The identification (ie. playlist name) can be either the playlist name, the concatenation of playlist name and directory path, or some other suitable name generation mechanism.

Browsing playlists without directory information:

AudioListInfo.StartResultAck(...,ResultData={playlistname}, FilterData={..PlaylistNameFilter= "*", MediaTypeFilter=b00001000}, ...)

The usage of the transport channel (Control/Packet Data) is determined by property AsyncControlSwitch. If set to True then all OPTypes will use the Packet Data Channel, if False then the Control Channel will be used.

2.1.23.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	SelectAudioListInfo (0x4A1)	StartResult Ack	SenderHandle , Start , NumberResults , ResultData , SortOrder1 , SortOrder2 , SortOrder3 , FilterData
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCode , ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , Start , NumberResults , ResultData , SortOrder1 , SortOrder2 , SortOrder3 , TotalNumberResults , AudioMediaData

2.1.23.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

Start

Start (together with NumberResults) indicates which part of the data is to be returned. The first record/line has a Start value of 1. A Start value greater than TotalNumberResults will result in an empty list.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

NumberResults

NumberResults indicates how many data entries are to be sent. Positive values are advancing in the result list whereas negative values are stepping backwards in the list. This criterion is applied last after applying SortOrder argument. Sending the special value of zero will return no entries, but will return a ResultAck message containing the TotalNumberResults to be expected for this request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

ResultData

ResultData indicates which record elements are of interest to the requester. Each bit represents one record element. Record elements which are not requested (i.e. the bit is set to false) are left out. Elements of the record which are not requested (i.e. the bit is set to false) are left out. Bit 0 has a special meaning. It can be used to switch over to Speech mode. If set, all String fields of AudioMediaData are then interpreted as phonetic data instead of the normal string data. Speech mode is optional and the FBlock should return an ErrorCode of 0x07 if speech mode has been requested but has not been implemented in the FBlock.

Basis datatype	Bit #	Code	Description
Unsigned Long	Bit 0	False	Normal (string) operation mode
		True	Speech mode for AudioMediaData parameter
	Bit 1	False	leave out
		True	Tag
	Bit 2	False	leave out
		True	HeaderInfoTitle
	Bit 3	False	leave out
		True	HeaderInfoArtist
	Bit 4	False	leave out
		True	HeaderInfoAlbum
	Bit 5	False	leave out
		True	HeaderInfoYear
	Bit 6	False	leave out
		True	HeaderInfoGenre
	Bit 7	False	leave out
		True	HeaderInfoComposer
	Bit 8	False	leave out
		True	HeaderInfoGrouping
	Bit 9	False	leave out
		True	HeaderInfoTrackNumber
	Bit 10	False	leave out
		True	HeaderInfoPlayTime
	Bit 11	False	leave out
		True	MediaFileFormat
	Bit 12	False	leave out
		True	HeaderInfoBitrate
	Bit 13	False	leave out
		True	HeaderInfoSamplerate
	Bit 14	False	leave out
		True	Filename
	Bit 15	False	leave out
		True	PlaylistName
	Bit 16	False	leave out
		True	DirectoryName
	Bit 17	False	leave out
		True	MediaType
Bit 18	False	leave out	
	True	AudioBook	

Basis datatype	Bit #	Code	Description
	Bit 19	False	leave out
		True	Podcast
	Bit 20	False	leave out
		True	Opus
	Bit 21	False	leave out
		True	Soloist
	Bit 22	False	leave out
		True	Conductor
	Bit 23	False	leave out
		True	Ensemble
	Bit 24	False	leave out
		True	NumberOfCoverArts

SortOrder1

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	none

SortOrder2

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	none

SortOrder3

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	none

FilterData

Please see the introduction of AuxIn concepts at the beginning of the document for the description of FilterData.

Basis datatype	Length	Condition	Description
Stream	3829	-	Content: TagFilter , HeaderInfoTitleFilter , HeaderInfoArtistFilter , HeaderInfoAlbumFilter , HeaderInfoYearFilter , HeaderInfoGenreFilter , HeaderInfoComposerFilter , HeaderInfoGroupingFilter , HeaderInfoTrackNumberFilter , HeaderInfoPlayTimeFilter , HeaderInfoBitrateFilter , HeaderInfoSamplerateFilter , FilenameFilter , PlaylistNameFilter , DirectoryNameFilter , MediaTypeFilter , AudioBookFilter , PodcastFilter , OpusFilter , SoloistFilter , ConductorFilter , EnsembleFilter , BoolOp

TagFilter

The Tag is a unique identifier for this entry. By giving Tag to MediaPosition, the AuxIn-FBlock must be able to uniquely identify which song to play.

Basis datatype	MaxSize
String	255

HeaderInfoTitleFilter

HeaderInfo* is information which can be taken directly from ID3-Tags.

Basis datatype	MaxSize
String	255

HeaderInfoArtistFilter

Basis datatype	MaxSize
String	255

HeaderInfoAlbumFilter

Basis datatype	MaxSize
String	255

HeaderInfoYearFilter

Basis datatype	MaxSize
String	255

HeaderInfoGenreFilter

Basis datatype	MaxSize
String	255

HeaderInfoComposerFilter

Basis datatype	MaxSize
String	255

HeaderInfoGroupingFilter

Basis datatype	MaxSize
String	255

HeaderInfoTrackNumberFilter

Basis datatype	MaxSize
String	255

HeaderInfoPlayTimeFilter

Basis datatype	MaxSize
String	255

HeaderInfoBitrateFilter

A positive value indicates a constant bitrate (CBR) as given. A negative value indicates a variable bitrate (VBR), averaging at the given bitrate.

Basis datatype	MaxSize
String	255

HeaderInfoSamplerateFilter

Basis datatype	MaxSize
String	255

FilenameFilter

Basis datatype	MaxSize
String	255

PlaylistNameFilter

Basis datatype	MaxSize
String	255

DirectoryNameFilter

A match for every directory can be accomplished with "/"

Basis datatype	MaxSize
String	255

MediaTypeFilter

Entry describes the requested type of the media. If more than one type is requested, each necessary bit has to be set to one.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	-
		True	Normal Audio
	Bit 1	False	-
		True	Directory
	Bit 2	False	-
		True	Category
	Bit 3	False	-
		True	Playlist
	Bit 4	False	-
		True	Audiobook
	Bit 5	False	-
		True	Podcast
	Bit 6	False	-

Basis datatype	Bit #	Code	Description
	Bit 7	True	FilesInPlaylist
		False	-
		True	Video

AudioBookFilter

Basis datatype	MaxSize
String	255

PodcastFilter

Basis datatype	MaxSize
String	255

OpusFilter

Basis datatype	MaxSize
String	255

SoloistFilter

Basis datatype	MaxSize
String	255

ConductorFilter

Basis datatype	MaxSize
String	255

EnsembleFilter

Basis datatype	MaxSize
String	255

BoolOp

BoolOp is a meta-tag which indicates how to combine the array elements when filtering. With regards to evaluation, AND takes precedence over OR. For the last record of the array, the BoolOp is ignored.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	OR
		True	AND
	Bit 1	False	Filter Not Valid
		True	Filter Valid

TotalNumberResults

The total number of results available which match the selected filter criteria. If the number cannot be determined at the time of the request, use 0xFFFF FFFF.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	0..0xFFFFFFFF	1	none

AudioMediaData

Due to the multitude of stream cases, there is no explicit list of these stream cases. Instead, only the most generic use case listing all possible parameters once within the stream case is shown. Depending on the Value of the Parameter 'ResultData', all data field which are NOT requested are left out.

The parameter sequence order is defined as follows and must be kept even though elements are left out:

Tag, HeaderInfoTitle, HeaderInfoArtist, HeaderInfoAlbum, HeaderInfoYear, HeaderInfoGenre, HeaderInfoComposer, HeaderInfoGrouping, HeaderInfoTrackNumber, HeaderInfoPlayTime, MediaFileFormat, HeaderInfoBitrate, HeaderInfoSamplerate, Filename, PlaylistName, DirectoryName, MediaType.

The reduced parameter set is repeated 'NumberResults' times to accommodate all requested result data.

All Examples: Assume Start is set to 4, NumberResults is set to 3,

Example 1: ResultData is 0x4 (i.e. Title is requested): Then, AudioMediaData will contain 'Title4,Title5,Title6'.

Example 2: ResultData is 0x1C (i.e. Title, Artist and Album are requested): Then, AudioMediaData will contain 'Title4,Artist4,Album4,Title5,Artist5,Album5,Title6,Artist6,Album6'

Basis datatype	Length	Condition	Description
Stream		-	Content: Tag , HeaderInfoTitle , HeaderInfoArtist , HeaderInfoAlbum , HeaderInfoYear , HeaderInfoGenre , HeaderInfoComposer , HeaderInfoGrouping , HeaderInfoTrackNumber , HeaderInfoPlayTime , MediaFileFormat , HeaderInfoBitrate , HeaderInfoSamplerate , Filename , Playlistname , Directoryname , MediaType , HeaderInfoAudioBook , HeaderInfoPodcast , HeaderInfoOpus , HeaderInfoSoloist , HeaderInfoConductor , HeaderInfoEnsemble , NumberOfCoverArts

Tag

The Tag is a unique identifier for this entry. By giving Tag to MediaPosition, the AuxIn-FBlock must be able to uniquely identify which song to play.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

HeaderInfoTitle

Basis datatype	MaxSize
String	255

HeaderInfoArtist

Basis datatype	MaxSize
String	255

HeaderInfoAlbum

Basis datatype	MaxSize
String	255

HeaderInfoYear

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

HeaderInfoGenre

Basis datatype	MaxSize
String	255

HeaderInfoComposer

Basis datatype	MaxSize
String	255

HeaderInfoGrouping

Basis datatype	MaxSize
String	255

HeaderInfoTrackNumber

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

HeaderInfoPlayTime

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

MediaFileFormat

Basis datatype	Range of values	Code	Description
Enum	0x00..0x09	0x00	Unknown
		0x01	Compressed Audio
		0x02	MP3
		0x03	WMA
		0x04	AAC
		0x05	OGGVorbis
		0x06	ATRAC
		0x07	MP3Pro
		0x08	WAV
		0x09	reserved

HeaderInfoBitrate

A positive value indicates a constant bitrate (CBR) as given. A negative value indicates a variable bitrate (VBR), averaging at the given bitrate.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Word	0	full range	1	none

HeaderInfoSamplerate

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

Filename

Basis datatype	MaxSize
String	255

Playlistname

Basis datatype	MaxSize
String	255

Directoryname

The directory in which the media is stored. When directory is not known (e.g. database oriented player) then an empty string is returned.

Basis datatype	MaxSize
String	255

MediaType

Entry describes the type of the media. "NormalAudio" is set for every media which is not one of the other types.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	-
		True	NormalAudio
	Bit 1	False	-
		True	Directory
	Bit 2	False	-
		True	Category
	Bit 3	False	-
		True	Playlist
	Bit 4	False	-
		True	Audiobook
	Bit 5	False	-
		True	Podcast
	Bit 6	False	-
		True	FilesInPlaylist
Bit 7	False	-	
	True	Video	

HeaderInfoAudioBook

Basis datatype	MaxSize
String	255

HeaderInfoPodcast

Basis datatype	MaxSize
String	255

HeaderInfoOpus

Basis datatype	MaxSize
String	255

HeaderInfoSoloist

Basis datatype	MaxSize
String	255

HeaderInfoConductor

Basis datatype	MaxSize
String	255

HeaderInfoEnsemble

Basis datatype	MaxSize
String	255

NumberOfCoverArts

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

2.1.24 SelectAudioListFilter (0x4A2)

Section Type: Extension

Set a Filter for the list of titles to play. Use AudioListInfo with the same filter to receive information about the number of titles selected by the filter and about the titles themselves. If no sort order is set, the sort order of the titles within the filtered result list is determined by a player internal mechanism. If a sort order has been set, then the ordering of the titles which are equal according to the given sort order is determined by a player internal mechanism.

When Database Navigation and AudioListFilter are used, the function TrackPosition is used to enumerate the titles which have been selected through AudioListFilter. AudioListFilterNotify function can be used to query the actual Filter and get notification on it.

This method does not affect the DeckStatus property. The controller should use other functions like DeckStatus, TrackPosition to achieve required HMI functionality.

2.1.24.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	SelectAudioList Filter (0x4A2)	StartResult Ack	SenderHandle , SortOrder1 , SortOrder2 , SortOrder3 , FilterData
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCode , ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle

2.1.24.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

SortOrder1

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	none

SortOrder2

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	none

SortOrder3

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	none

FilterData

Please see the introduction of AuxIn concepts at the beginning of the document for the description of FilterData.

Basis datatype	Length	Condition	Description
Stream	3829	-	Content: TagFilter , HeaderInfoTitleFilter , HeaderInfoArtistFilter , HeaderInfoAlbumFilter , HeaderInfoYearFilter , HeaderInfoGenreFilter , HeaderInfoComposerFilter , HeaderInfoGroupingFilter , HeaderInfoTrackNumberFilter , HeaderInfoPlayTimeFilter , HeaderInfoBitrateFilter , HeaderInfoSamplerateFilter , FilenameFilter , PlaylistNameFilter , DirectoryNameFilter , MediaTypeFilter , AudioBookFilter , PodcastFilter , OpusFilter , SoloistFilter , ConductorFilter , EnsembleFilter , BoolOp

TagFilter

The Tag is a unique identifier for this entry. By giving Tag to MediaPosition, the AuxIn-FBlock must be able to uniquely identify which song to play.

Basis datatype	MaxSize
String	255

HeaderInfoTitleFilter

HeaderInfo* is information which can be taken directly from ID3-Tags.

Basis datatype	MaxSize
String	255

HeaderInfoArtistFilter

Basis datatype	MaxSize
String	255

HeaderInfoAlbumFilter

Basis datatype	MaxSize
String	255

HeaderInfoYearFilter

Basis datatype	MaxSize
String	255

HeaderInfoGenreFilter

Basis datatype	MaxSize
String	255

HeaderInfoComposerFilter

Basis datatype	MaxSize
String	255

HeaderInfoGroupingFilter

Basis datatype	MaxSize
String	255

HeaderInfoTrackNumberFilter

Basis datatype	MaxSize
String	255

HeaderInfoPlayTimeFilter

Basis datatype	MaxSize
String	255

HeaderInfoBitrateFilter

A positive value indicates a constant bitrate (CBR) as given. A negative value indicates a variable bitrate (VBR), averaging at the given bitrate.

Basis datatype	MaxSize
String	255

HeaderInfoSamplerateFilter

Basis datatype	MaxSize
String	255

FilenameFilter

Basis datatype	MaxSize
String	255

PlaylistNameFilter

Basis datatype	MaxSize
String	255

DirectoryNameFilter

A match for every directory can be accomplished with "/"

Basis datatype	MaxSize
String	255

MediaTypeFilter

Entry describes the requested type of the media. If more than one type is requested, each necessary bit has to be set to one.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	-
		True	Normal Audio
	Bit 1	False	-
		True	Directory
	Bit 2	False	-
		True	Category
	Bit 3	False	-
		True	Playlist
	Bit 4	False	-
		True	Audiobook
	Bit 5	False	-
		True	Podcast
	Bit 6	False	-
		True	FilesInPlaylist
	Bit 7	False	-
		True	Video

AudioBookFilter

Basis datatype	MaxSize
String	255

PodcastFilter

Basis datatype	MaxSize
String	255

OpusFilter

Basis datatype	MaxSize
String	255

SoloistFilter

Basis datatype	MaxSize
String	255

ConductorFilter

Basis datatype	MaxSize
String	255

EnsembleFilter

Basis datatype	MaxSize
String	255

BoolOp

BoolOp is a meta-tag which indicates how to combine the array elements when filtering. With regards to evaluation, AND takes precedence over OR. For the last record of the array, the BoolOp is ignored.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	OR
		True	AND
	Bit 1	False	Filter Not Valid
		True	Filter Valid

2.1.25 CurrentAudioListFilter (0x4A3)

Section Type: Extension

Property which is used by the controller to query the current AudioFilter which is used for playback. Setting the filter is done using SelectAudioListFilter method. An unknown filter is signaled with BoolOp value.

Notification may be possible.

2.1.25.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
AuxIn (0x24)	CurrentAudioList Filter (0x4A3)	Get	-
		Status	SortOrder1 , SortOrder2 , SortOrder3 , FilterData
		Error	ErrorCode, ErrorInfo

2.1.25.2 Parameter

SortOrder1

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	none

SortOrder2

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	none

SortOrder3

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	full range	1	none

FilterData

Please see the introduction of AuxIn concepts at the beginning of the document for the description of FilterData.

Basis datatype	Length	Condition	Description
Stream	3829	-	Content: TagFilter , HeaderInfoTitleFilter , HeaderInfoArtistFilter , HeaderInfoAlbumFilter , HeaderInfoYearFilter , HeaderInfoGenreFilter , HeaderInfoComposerFilter , HeaderInfoGroupingFilter , HeaderInfoTrackNumberFilter , HeaderInfoPlayTimeFilter , HeaderInfoBitrateFilter , HeaderInfoSamplerateFilter , FilenameFilter , PlaylistNameFilter , DirectoryNameFilter , MediaTypeFilter , AudioBookFilter , PodcastFilter , OpusFilter , SoloistFilter , ConductorFilter , EnsembleFilter , BoolOp

TagFilter

The Tag is a unique identifier for this entry. By giving Tag to MediaPosition, the AuxIn-FBlock must be able to uniquely identify which song to play.

Basis datatype	MaxSize
String	255

HeaderInfoTitleFilter

HeaderInfo* is information which can be taken directly from ID3-Tags.

Basis datatype	MaxSize
String	255

HeaderInfoArtistFilter

Basis datatype	MaxSize
String	255

HeaderInfoAlbumFilter

Basis datatype	MaxSize
String	255

HeaderInfoYearFilter

Basis datatype	MaxSize
String	255

HeaderInfoGenreFilter

Basis datatype	MaxSize
String	255

HeaderInfoComposerFilter

Basis datatype	MaxSize
String	255

HeaderInfoGroupingFilter

Basis datatype	MaxSize
String	255

HeaderInfoTrackNumberFilter

Basis datatype	MaxSize
String	255

HeaderInfoPlayTimeFilter

Basis datatype	MaxSize
String	255

HeaderInfoBitrateFilter

A positive value indicates a constant bitrate (CBR) as given. A negative value indicates a variable bitrate (VBR), averaging at the given bitrate.

Basis datatype	MaxSize
String	255

HeaderInfoSamplerateFilter

Basis datatype	MaxSize
String	255

FilenameFilter

Basis datatype	MaxSize
String	255

PlaylistNameFilter

Basis datatype	MaxSize
String	255

DirectoryNameFilter

A match for every directory can be accomplished with "/"

Basis datatype	MaxSize
String	255

MediaTypeFilter

Entry describes the requested type of the media. If more than one type is requested, each necessary bit has to be set to one.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	-
		True	Normal Audio
	Bit 1	False	-
		True	Directory
	Bit 2	False	-
		True	Category
	Bit 3	False	-
		True	Playlist
	Bit 4	False	-
		True	Audiobook
	Bit 5	False	-
		True	Podcast
	Bit 6	False	-
		True	FilesInPlaylist
Bit 7	False	-	
	True	Video	

AudioBookFilter

Basis datatype	MaxSize
String	255

PodcastFilter

Basis datatype	MaxSize
String	255

OpusFilter

Basis datatype	MaxSize
String	255

SoloistFilter

Basis datatype	MaxSize
String	255

ConductorFilter

Basis datatype	MaxSize
String	255

EnsembleFilter

Basis datatype	MaxSize
String	255

BoolOp

BoolOp is a meta-tag which indicates how to combine the array elements when filtering. With regards to evaluation, AND takes precedence over OR. For the last record of the array, the BoolOp is ignored.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	OR
		True	AND
	Bit 1	False	Filter Not Valid
		True	Filter Valid

2.1.26 SelectCurrentAudioListInfo (0x4A4)

Section Type: Extension

Get audio specific information from a (possibly virtual) audio media information database on the AuxIn FBlock or on the device represented by the AuxIn FBlock. The returned list contains a thesaurus of audio specific information of all audio media available. This function behaves the same as AudioListInfo with one difference that it does not use a FilterData argument but rather queries the currently selected tracklist of the external device.

Notes:

- The record used for the AudioMediaData parameter can be considered as a (non-optimized, redundant) table in a relational database
- If the filter is set and produces duplicate entries, the duplicates must be removed before transfer
- Record elements which are not requested should be empty (string: empty string, numbers: zero)
- It is advised to set the ResultData bit for PlayList with caution. Requesting all PlayLists with all Songs can return a huge number of result records: If a Song is in more than one PlayList, it will be returned once for each PlayList it is in (RIGHT OUTER JOIN operation between playlists and songs).
- If NumberResults is a large value and the FBlock does not have enough memory available to process the request, use ErrorCode 0x20, ErrorInfo 0x02 (List overflow) to indicate that the FBlock is not able to handle this request.

The usage of the transport channel (Control/Packet Data) is determined by property AsyncControlSwitch. If set to True then all OPTypes will use the Packet Data Channel, if False then the Control Channel will be used.

2.1.26.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	SelectCurrentAudioListInfo (0x4A4)	StartResult	SenderHandle , Start , Offset ,
		Ack	NumberResults , ResultData
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCode , ErrorInfo
		ProcessingAck	SenderHandle
	ResultAck	SenderHandle , Start , Offset ,	
			NumberResults , ResultData , TotalNumberResults , AudioMediaData

2.1.26.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

Start

Start (together with Offset and NumberResults) indicates which part of the data is to be returned. The first record/line has a Start value of 1.

If Start is set to 0 (zero), information about the current title is requested without the necessity for additional knowledge of the current TrackPosition or MediaPosition/Tag.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

Offset

This offset is added/subtracted to the Start parameter in order to set the start of the window even if TrackPosition is not known, i.e. Start = 0.

Example:

Start=0

Offset=10

NumberResults=5

This will query in total the meta data of 5 entries which are 10 positions ahead of the current play position.

Start=0

Offset=-10

NumberResults=5

This will query in total the meta data of 5 entries which are starting 10 positions behind of the current play position (-10 ... -6).

This function can be conveniently used to maintain a sliding window of cached metadata around the currently playing track (even if TrackPosition is not known)

Basis datatype	Exp.	Range of values	Step	Unit
Signed Word	0	full range	1	none

NumberResults

NumberResults indicates how many data entries are to be sent. Sending the special value of zero will return no entries, but will return a ResultAck message containing the TotalNumberResults to be expected for this request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

ResultData

ResultData indicates which record elements are of interest to the requester. Each bit represents one record element. Record elements which are not requested (i.e. the bit is set to false) are left out. Elements of the record which are not requested (i.e. the bit is set to false) are left out. Bit 0 has a special meaning. It can be used to switch over to Speech mode. If set, all String fields of AudioMediaData are then interpreted as phonetic data instead of the normal string data. Speech mode is optional and the FBlock should return an ErrorCode of 0x07 if speech mode has been requested but has not been implemented in the FBlock.

Basis datatype	Bit #	Code	Description
Unsigned Long	Bit 0	False	Normal (string) operation mode
		True	Speech mode for AudioMediaData parameter
	Bit 1	False	leave out
		True	Tag
	Bit 2	False	leave out
		True	HeaderInfoTitle
	Bit 3	False	leave out
		True	HeaderInfoArtist
	Bit 4	False	leave out
		True	HeaderInfoAlbum
	Bit 5	False	leave out
		True	HeaderInfoYear
	Bit 6	False	leave out
		True	HeaderInfoGenre
	Bit 7	False	leave out
		True	HeaderInfoComposer
	Bit 8	False	leave out
		True	HeaderInfoGrouping
	Bit 9	False	leave out
		True	HeaderInfoTrackNumber
	Bit 10	False	leave out
		True	HeaderInfoPlayTime
	Bit 11	False	leave out
		True	MediaFileFormat
	Bit 12	False	leave out
		True	HeaderInfoBitrate
	Bit 13	False	leave out
		True	HeaderInfoSamplerate
	Bit 14	False	leave out
		True	Filename
	Bit 15	False	leave out
		True	PlaylistName
	Bit 16	False	leave out
		True	DirectoryName
	Bit 17	False	leave out
		True	MediaType
	Bit 18	False	leave out
		True	AudioBook
	Bit 19	False	leave out
		True	Podcast
	Bit 20	False	leave out
		True	Opus
	Bit 21	False	leave out
		True	Soloist
	Bit 22	False	leave out
		True	Conductor
	Bit 23	False	leave out
		True	Ensemble
Bit 24	False	leave out	
	True	NumberOfCoverArts	

TotalNumberResults

The total number of results available which match the selected filter criteria. If the number cannot be determined at the time of the request, use 0xFFFF FFFF.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	0..0xFFFFFFFF	1	none

AudioMediaData

Due to the multitude of stream cases, there is no explicit list of these stream cases. Instead, only the most generic use case listing all possible parameters once within the stream case is shown. Depending on the Value of the Parameter 'ResultData', all data field which are NOT requested are left out.

The parameter sequence order is defined as follows and must be kept even though elements are left out:

Tag, HeaderInfoTitle, HeaderInfoArtist, HeaderInfoAlbum, HeaderInfoYear, HeaderInfoGenre, HeaderInfoComposer, HeaderInfoGrouping, HeaderInfoTrackNumber, HeaderInfoPlayTime, MediaFileFormat, HeaderInfoBitrate, HeaderInfoSamplerate, Filename, PlaylistName, DirectoryName, MediaType.

The reduced parameter set is repeated 'NumberResults' times to accommodate all requested result data.

All Examples: Assume Start is set to 4, NumberResults is set to 3,

Example 1: ResultData is 0x4 (i.e. Title is requested): Then, AudioMediaData will contain 'Title4,Title5,Title6'.

Example 2: ResultData is 0x1C (i.e. Title, Artist and Album are requested): Then, AudioMediaData will contain 'Title4,Artist4,Album4,Title5,Artist5,Album5,Title6,Artist6,Album6'

Basis datatype	Length	Condition	Description
Stream		-	Content: Tag , HeaderInfoTitle , HeaderInfoArtist , HeaderInfoAlbum , HeaderInfoYear , HeaderInfoGenre , HeaderInfoComposer , HeaderInfoGrouping , HeaderInfoTrackNumber , HeaderInfoPlayTime , MediaFileFormat , HeaderInfoBitrate , HeaderInfoSamplerate , Filename , Playlistname , Directoryname , MediaType , HeaderInfoAudioBook , HeaderInfoPodcast , HeaderInfoOpus , HeaderInfoSoloist , HeaderInfoConductor , HeaderInfoEnsemble , NumberOfCoverArts

Tag

The Tag is a unique identifier for this entry. By giving Tag to MediaPosition, the AuxIn-FBlock must be able to uniquely identify which song to play.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

HeaderInfoTitle

Basis datatype	MaxSize
String	255

HeaderInfoArtist

Basis datatype	MaxSize
String	255

HeaderInfoAlbum

Basis datatype	MaxSize
String	255

HeaderInfoYear

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

HeaderInfoGenre

Basis datatype	MaxSize
String	255

HeaderInfoComposer

Basis datatype	MaxSize
String	255

HeaderInfoGrouping

Basis datatype	MaxSize
String	255

HeaderInfoTrackNumber

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

HeaderInfoPlayTime

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

MediaFileFormat

Basis datatype	Range of values	Code	Description
Enum	0x00..0x09	0x00	Unknown
		0x01	Compressed Audio
		0x02	MP3
		0x03	WMA
		0x04	AAC
		0x05	OGGVorbis
		0x06	ATRAC
		0x07	MP3Pro
		0x08	WAV
		0x09	reserved

HeaderInfoBitrate

A positive value indicates a constant bitrate (CBR) as given. A negative value indicates a variable bitrate (VBR), averaging at the given bitrate.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Word	0	full range	1	none

HeaderInfoSamplerate

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

Filename

Basis datatype	MaxSize
String	255

Playlistname

Basis datatype	MaxSize
String	255

Directoryname

The directory in which the media is stored. When directory is not known (e.g. database oriented player) then an empty string is returned.

Basis datatype	MaxSize
String	255

MediaType

Entry describes the type of the media. "NormalAudio" is set for every media which is not one of the other types.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	-
		True	NormalAudio
	Bit 1	False	-
		True	Directory
	Bit 2	False	-
		True	Category
	Bit 3	False	-
		True	Playlist
	Bit 4	False	-
		True	Audiobook
	Bit 5	False	-
		True	Podcast
	Bit 6	False	-
		True	FilesInPlaylist
Bit 7	False	-	
	True	Video	

HeaderInfoAudioBook

Basis datatype	MaxSize
String	255

HeaderInfoPodcast

Basis datatype	MaxSize
String	255

HeaderInfoOpus

Basis datatype	MaxSize
String	255

HeaderInfoSoloist

Basis datatype	MaxSize
String	255

HeaderInfoConductor

Basis datatype	MaxSize
String	255

HeaderInfoEnsemble

Basis datatype	MaxSize
String	255

NumberOfCoverArts

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

2.1.27 CurrentMediaListContentChanged (0x4A5)

Section Type: Extension

If the playlist is updated due to content changes, the value of this property is changed to indicate that the cache should be invalidated.

CurrentMediaListContentChanged has to support notification.

2.1.27.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
AuxIn (0x24)	CurrentMediaList ContentChanged (0x4A5)	Get	-
		Status	Counter
		Error	ErrorCode, ErrorInfo

2.1.27.2 Parameter

Counter

The current value of the CurrentMediaListContentChanged property.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	-		-	none

2.1.28 SelectCoverArt (0x4A6)

Section Type: Extension

Get cover art specific raw data from a media information database on the AuxIn FBlock or on the device represented by the AuxIn FBlock. The response data contains a list of the corresponding cover arts, which can be matched to the requested audio files via the parameter TagLong.

Notes:

- The transmission of SelectCoverArt.ResultAck is done via the MOST High Protocol (Async Data Transmission Service, Block Acknowledge).
- The transmission of SelectCoverArt.StartResultAck, SelectCoverArt.AbortAck, SelectCoverArt.ProcessingAck and SelectCoverArt.ErrorAck is done via the Control Channel

2.1.28.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	SelectCoverArt (0x4A6)	StartResult Ack	SenderHandle , Start , NumberResults , CoverArtSizeID , CoverResultData , SortOrder1 , SortOrder2 , SortOrder3 , FilterData
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCode , ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , Start , NumberResults , CoverArtSizeID , CoverResultData , TotalNumberResults , TotalNumberOfCovers , CoverArtData

2.1.28.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

Start

Start (together with NumberResults) indicates which part of the data is to be returned. The first record/line has a Start value of 1. A Start value greater than TotalNumberResults will result in an empty list. If Start is set to 0 (zero), information about the current title is requested without the necessity for additional knowledge of the current TrackPosition or MediaPosition/Tag.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	none

NumberResults

NumberResults indicates how many data entries are to be sent. Positive values are advancing in the result list whereas negative values are stepping backwards in the list. This criterion is applied last after applying SortOrder argument. Sending the special value of zero will return no entries, but will return a ResultAck message containing the TotalNumberResults to be expected for this request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

CoverArtSizeID

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	original size
		0x01	very small
		0x02	small
		0x03	medium
		0x04	large
		0x05	very large

CoverResultData

CoverResultData indicates which record elements are of interest to the requester. Each bit represents one record element. Elements of the record which are `_NOT_` requested (i.e. the bit is set to false) are left out.

Notes:

- To match the cover arts to an audio track list which is requested via the function `SelectAudioListInfo`, the Bit 0 (`TagLong`) is set to TRUE.
- If the raw data of the cover art (`CoverArtRawData`) is requested, the bit for `CoverArtRawDataLength` is also set to TRUE.

Basis datatype	Bit #	Code	Description
Unsigned Word	Bit 0	False	leave out
		True	TagLong
	Bit 1	False	leave out
		True	CoverArtType
	Bit 2	False	leave out
		True	CoverArtFileFormat
	Bit 3	False	leave out
		True	CoverArtRawDataLength
	Bit 4	False	leave out
		True	CoverArtRawData
	Bit 5	False	leave out
		True	reserved
	Bit 6	False	leave out
		True	reserved
	Bit 7	False	leave out
		True	reserved
	Bit 8	False	leave out
		True	reserved
	Bit 9	False	leave out
		True	reserved
Bit 10	False	leave out	
	True	reserved	

Basis datatype	Bit #	Code	Description
	Bit 11	False	leave out
		True	reserved
	Bit 12	False	leave out
		True	reserved
	Bit 13	False	leave out
		True	reserved
	Bit 14	False	leave out
		True	reserved
Bit 15	False	leave out	
	True	reserved	

SortOrder1

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	-128..127	1	none

SortOrder2

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	-128..127	1	none

SortOrder3

Please see the introduction of AuxIn concepts at the beginning of the document for the description of the SortOrder parameters.

Basis datatype	Exp.	Range of values	Step	Unit
Signed Byte	0	-128..127	1	none

FilterData

Please see the introduction of AuxIn concepts at the beginning of the document for the description of FilterData.

Basis datatype	Length	Condition	Description
Stream	3829	-	Content: TagFilter , HeaderInfoTitleFilter , HeaderInfoArtistFilter , HeaderInfoAlbumFilter , HeaderInfoYearFilter , HeaderInfoGenreFilter , HeaderInfoComposerFilter , HeaderInfoGroupingFilter , HeaderInfoTrackNumberFilter , HeaderInfoPlayTimeFilter , HeaderInfoBitrateFilter , HeaderInfoSamplerateFilter , FilenameFilter , PlaylistNameFilter , DirectoryNameFilter , MediaTypeFilter , AudioBookFilter , PodcastFilter , OpusFilter , SoloistFilter , ConductorFilter , EnsembleFilter , BoolOp

TagFilter

The Tag is a unique identifier for this entry. By giving Tag to MediaPosition, the AuxIn-FBlock must be able to uniquely identify which song to play.

Basis datatype	MaxSize
String	255

HeaderInfoTitleFilter

HeaderInfo* is information which can be taken directly from ID3-Tags.

Basis datatype	MaxSize
String	255

HeaderInfoArtistFilter

Basis datatype	MaxSize
String	255

HeaderInfoAlbumFilter

Basis datatype	MaxSize
String	255

HeaderInfoYearFilter

Basis datatype	MaxSize
String	255

HeaderInfoGenreFilter

Basis datatype	MaxSize
String	255

HeaderInfoComposerFilter

Basis datatype	MaxSize
String	255

HeaderInfoGroupingFilter

Basis datatype	MaxSize
String	255

HeaderInfoTrackNumberFilter

Basis datatype	MaxSize
String	255

HeaderInfoPlayTimeFilter

Basis datatype	MaxSize
String	255

HeaderInfoBitrateFilter

A positive value indicates a constant bitrate (CBR) as given. A negative value indicates a variable bitrate (VBR), averaging at the given bitrate.

Basis datatype	MaxSize
String	255

HeaderInfoSamplerateFilter

Basis datatype	MaxSize
String	255

FilenameFilter

Basis datatype	MaxSize
String	255

PlaylistNameFilter

Basis datatype	MaxSize
String	255

DirectoryNameFilter

A match for every directory can be accomplished with "/"*

Basis datatype	MaxSize
String	255

MediaTypeFilter

Entry describes the requested type of the media. If more than one type is requested, each necessary bit has to be set to one.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	-
		True	Normal Audio
	Bit 1	False	-
		True	Directory
	Bit 2	False	-
		True	Category
	Bit 3	False	-
		True	Playlist
	Bit 4	False	-
		True	Audiobook
	Bit 5	False	-
		True	Podcast
	Bit 6	False	-

Basis datatype	Bit #	Code	Description
	Bit 7	True	FilesInPlaylist
		False	-
		True	Video

AudioBookFilter

Basis datatype	MaxSize
String	255

PodcastFilter

Basis datatype	MaxSize
String	255

OpusFilter

Basis datatype	MaxSize
String	255

SoloistFilter

Basis datatype	MaxSize
String	255

ConductorFilter

Basis datatype	MaxSize
String	255

EnsembleFilter

Basis datatype	MaxSize
String	255

BoolOp

BoolOp is a meta-tag which indicates how to combine the array elements when filtering. With regards to evaluation, AND takes precedence over OR. For the last record of the array, the BoolOp is ignored.

Basis datatype	Bit #	Code	Description
Unsigned Byte	Bit 0	False	OR
		True	AND
	Bit 1	False	Filter Not Valid
		True	Filter Valid

Basis datatype	Range of values	Code	Description
		0x0B	Composer
		0x0C	Lyricist/text writer
		0x0D	Recording Location
		0x0E	During recording
		0x0F	During performance
		0x10	Movie/video screen capture
		0x11	A bright coloured fish
		0x12	Illustration
		0x13	Band/artist logotype
		0x14	Publisher/Studio logotype

CoverArtFileFormat

Basis datatype	Range of values	Code	Description
Enum	0x00..0x06	0x00	Unknown
		0x01	JPEG
		0x02	PNG
		0x03	GIF
		0x04	Icon
		0x05	TIFF
		0x06	Bitmap

CoverArtRawDataLength

Length of the following data stream.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	0..0xFFFFFFFF	1	none

CoverArtRawData

The actual raw picture data.

Basis datatype	Length	Condition	Description
Stream		-	

2.1.29 SelectCurrentCoverArt (0x4A7)

Section Type: Extension

Get cover art specific raw data from a media information database on the AuxIn FBlock or on the device represented by the AuxIn FBlock. The response data contains a list of the corresponding cover arts, which can be matched to the requested audio files via the parameter TagLong. This function behaves the same as SelectCoverArt with one difference that it does not use a FilterData argument but rather queries the currently selected tracklist of the external device.

Notes:

- The record used for the CoverArtData parameter can be considered as a (non-optimized, redundant) table in a relational database.
- If the filter is set and produces duplicate entries, the duplicates must be removed before transfer.
- Record elements which are not requested should be empty (string: empty string, numbers: zero) .
- More than one CoverArt per title can be returned.
- If NumberResults is a large value and the FBlock does not have enough memory available to process the request, use ErrorCode 0x20, ErrorInfo 0x02 (List overflow) to indicate that the FBlock is not able to handle this request.
- The transmission of SelectCurrentCoverArt.ResultAck should be done via the MOST High Protocol (Async Data Transmission Service, Block Acknowledge).
- The transmission of SelectCurrentCoverArt.StartResultAck, SelectCurrentCoverArt.AbortAck, SelectCurrentCoverArt.ProcessingAck and SelectCurrentCoverArt.ErrorAck is done via the Control Channel.

2.1.29.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	SelectCurrentCoverArt (0x4A7)	StartResultAck	SenderHandle , Start , StartOffset , NumberResults , CoverArtSizeID , CoverResultData
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCode , ErrorInfo
		ProcessingAck	SenderHandle
		ResultAck	SenderHandle , Start , StartOffset , NumberResults , CoverArtSizeID , CoverResultData , TotalNumberResults , TotalNumberOfCovers , CoverArtData

2.1.29.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

Start

Start (together with NumberResults) indicates which part of the data is to be returned. The first record/line has a Start value of 1. A Start value greater than TotalNumberResults will result in an empty list. If Start is set to 0 (zero), information about the current title is requested without the necessity for additional knowledge of the current TrackPosition or MediaPosition/Tag.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	0..0xFFFFFFFF	1	none

StartOffset

This offset is added to or subtracted from the Start parameter in order to set the start of the window even if TrackPosition is not known, i.e., Start = 0.

Example:

Start=0

Offset=10

NumberResults=5

This will query in total the meta data of 5 entries which are 10 positions ahead of the current play position.

Start=0

Offset=-10

NumberResults=5

This will query in total the meta data of 5 entries which are starting 10 positions behind of the current play position (-10 ... -6).

This function can be conveniently used to maintain a sliding window of cached metadata around the currently playing track (even if TrackPosition is not known).

Basis datatype	Exp.	Range of values	Step	Unit
Signed Word	0	-32768..32767	1	none

NumberResults

NumberResults indicates how many data entries are to be sent. Positive values are advancing in the result list whereas negative values are stepping backwards in the list. This criterion is applied last after applying SortOrder argument. Sending the special value of zero will return no entries, but will return a ResultAck message containing the TotalNumberResults to be expected for this request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

CoverArtSizeID

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	original size
		0x01	very small
		0x02	small
		0x03	medium
		0x04	large
		0x05	very large

CoverResultData

CoverResultData indicates which record elements are of interest to the requester. Each bit represents one record element. Elements of the record which are NOT requested (i.e. the bit is set to false) are left out.

Notes:

- To match the cover arts to an audio track list which is requested via the function SelectAudioListInfo, the Bit 0 (TagLong) is set to TRUE.
- If the raw data of the cover art (CoverArtData) is requested, the bit for CoverArtRawDataLength is also set to TRUE.

Basis datatype	Bit #	Code	Description
Unsigned Word	Bit 0	False	leave out
		True	TagLong
	Bit 1	False	leave out
		True	CoverArtType
	Bit 2	False	leave out
		True	CoverArtFileFormat
	Bit 3	False	leave out
		True	CoverArtRawDataLength
	Bit 4	False	leave out
		True	CoverArtData
	Bit 5	False	leave out
		True	reserved
	Bit 6	False	leave out
		True	reserved
	Bit 7	False	leave out
		True	reserved
	Bit 8	False	leave out
		True	reserved
	Bit 9	False	leave out
		True	reserved
	Bit 10	False	leave out
		True	reserved
	Bit 11	False	leave out
		True	reserved
	Bit 12	False	leave out
		True	reserved
	Bit 13	False	leave out
		True	reserved
	Bit 14	False	leave out
		True	reserved
Bit 15	False	leave out	
	True	reserved	

TotalNumberResults

The total number of results available which match the selected filter criteria. If the total number of results is not known (e.g. total filter content could not be computed), then TotalNumberResults = 0xFFFF FFFF should be returned.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	0..0xFFFFFFFF	1	none

TotalNumberOfCovers

The total number of covers available. If the total number of covers is not known (e.g., total filter content could not be computed), then TotalNumberOfCovers = 0xFFFF FFFF is returned.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	0..0xFFFFFFFF	1	none

CoverArtData

Stream of cover art data. Due to the multitude of the stream cases, there is no explicit list of these stream cases. Depending on the value of the parameter 'CoverResultData', all data fields which are NOT requested are left out. For example, parameter Start is set to 4; parameter NumberResults is set to 3. In case CoverResultData is set to 0x3 (i.e. CoverArtRawDataLength is requested), then stream of CoverArtData will only contain 'CoverArtRawDataLength[4],CoverArtRawDataLength[5],CoverArtRawDataLength[6]'.

Basis datatype	Length	Condition	Description
Stream		-	Content: TagLong , CoverArtType , CoverArtFileType , CoverArtRawDataLength , CoverArtRawData {TagLong[1], CoverArtType[1], CoverArtFileType[1], CoverArtRawDataLength[1], CoverArtData[1];... ;TagLong[n], CoverArtType[n], CoverArtFileType[n], CoverArtRawDataLength[n], CoverArtData[n]}

TagLong

Unique identifier.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	0..0xFFFFFFFF	1	none

CoverArtType

Basis datatype	Range of values	Code	Description
Enum	0x0..0x14	0x0	Other
		0x1	32x32 pixels 'file icon' (PNG only)
		0x2	Other file icon
		0x3	Cover (front)
		0x4	Cover (back)
		0x5	Leaflet page
		0x6	Media (e.g. label side of CD)
		0x7	Lead artist/lead performer/soloist
		0x8	Artist/performer
		0x9	Conductor
		0xA	Band/Orchestra

Basis datatype	Range of values	Code	Description
		0xB	Composer
		0xC	Lyricist/text writer
		0xD	Recording Location
		0xE	During recording
		0xF	During performance
		0x10	Movie/video screen capture
		0x11	A bright coloured fish
		0x12	Illustration
		0x13	Band/artist logotype
		0x14	Publisher/Studio logotype

CoverArtFileFormat

Basis datatype	Range of values	Code	Description
Enum	0x0..0x6	0x0	Unknown
		0x1	JPEG
		0x2	PNG
		0x3	GIF
		0x4	Icon
		0x5	TIFF
		0x6	Bitmap

CoverArtRawDataLength

Length of the following data stream.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	0..0xFFFFFFFF	1	none

CoverArtRawData

The actual raw picture data.

Basis datatype	Length	Condition	Description
Stream		-	

2.1.30 CoverArtSetup (0x4A8)

Section Type: Extension

This function is used to set predefined sizes of pictures for the use with picture requests. Therefore PosX defines the CoverArtSizeID to request a picture in the according size. The CoverArtSizeID (PosX) of 0 cannot be set. It is used to request the original size of the picture. Example: The value of PosX,PosY [1,0] is set to Width=300, Height=200. When the picture is requested with a CoverArtSizeID of 1, the picture will be sent with a size of 300x200 pixels. When the picture is requested with a CoverArtSizeID of 0, the picture will be sent with its original size.

2.1.30.1 Format of Function

Function classes: Array of { Record of { Number Number Number } }

FBlock	Function	OPType	Parameter
AuxIn (0x24)	CoverArtSetup (0x4A8)	Set	Pos, Data
		Get	Pos
		SetGet	Pos, Data
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.30.2 Parameter

Pos

The parameter Pos={x,y} consists of two bytes, x and y, and shows which parameter shall be set, queried or read. Since this property has only one dimension, y is unused.
Valid range: x=1..5, y=0

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

Data

Basis datatype	Length	Description	
Array	-	Pos	Data
		{ x=0, y=0 }	Width[1], Height[1], QualityLevel [1], Width[2], Height[2], QualityLevel [2],..., Width[N], Height[N], QualityLevel [N]

Width

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

Height

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

QualityLevel

Indicates the quality in percent.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0..100	1	none

2.1.31 RetrieveCoverArt (0x4A9)

Section Type: Extension

This method requests a certain cover art for a given tag. The request may be split into small pieces, down to a single byte. The slave may send less data than requested. This is not an error if it happens for the final chunk.

2.1.31.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	RetrieveCoverArt (0x4A9)	StartResult	SenderHandle , TagLong , PictureIndex , CoverArtSizeID , Offset , Count
		Ack	
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCode , ErrorInfo
		Processing Ack	SenderHandle
	ResultAck	SenderHandle , TagLong , PictureIndex , CoverArtSizeID , Offset , Count , Data	

2.1.31.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

TagLong

TagLong is a unique identifier within the media library.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

PictureIndex

The current request asks for position PictureIndex within the ID3 tags specifying cover art. Which PictureIndex fits can be determined by looking at the parameter CoverArtData.CoverArtType of a previous call to SelectCoverArt.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

CoverArtSizeID

Basis datatype	Range of values	Code	Description
Enum	0x00..0x05	0x00	original size

Basis datatype	Range of values	Code	Description
		0x01	very small
		0x02	small
		0x03	medium
		0x04	large
		0x05	very large

Offset

Defines the starting byte for transfer. It is highly recommended to work with proper byte alignment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

Count

Defines the amount of bytes to transfer. It is highly recommended to work with proper byte alignment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

Data

The raw data is sent as stream of bytes.

Basis datatype	Length	Condition	Description
Stream		-	Contains Count number of bytes. No escaping of any characters.

2.1.32 SelectAuxPlayer (0x4AA)

Section Type: Extension

This function selects a certain AuxPlayer, which is referenced by the AuxPlayerID defined in function AvailableAuxPlayer.

2.1.32.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	SelectAuxPlayer (0x4AA)	StartResult Ack	SenderHandle , AuxPlayerID
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCode, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle

2.1.32.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

AuxPlayerID

Identifier of an AuxPlayer. Default value: 0

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

2.1.33 AvailableAuxPlayer (0x4AB)

Section Type: Extension

This function provides a list of all available AuxPlayer (software emulated players) on an external CE device. The CE device is connected for instance via USB interface or via Bluetooth using the A2DP profile.

Notes:

- The transmission of AvailableAuxPlayer.Status is done via the MOST High Protocol (Async Data Transmission Service, Block Acknowledge).
- The transmission of AvailableAuxPlayer.Get and AvailableAuxPlayer.Error is done via the Control Channel.

2.1.33.1 Format of Function

Function classes: Array of { Record of { Number String } }

FBlock	Function	OPType	Parameter
AuxIn (0x24)	AvailableAuxPlayer (0x4AB)	Get	Pos
		Status	Pos, Data
		Error	ErrorCode, ErrorInfo

2.1.33.2 Parameter

Pos

The parameter Pos={x,y} consists of two bytes, x and y, and shows which parameter shall be set, queried or read.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

Data

Basis datatype	Length	Description	
		Pos	Data
Array	-	{ x=0, y=0 }	AuxPlayerID[1], AuxPlayerName[1], AuxPlayerID[2], AuxPlayerName[2],..., AuxPlayerID[N], AuxPlayerName[N]
		{ x=x, y=0 }	AuxPlayerID[x], AuxPlayerName[x]

AuxPlayerID

Identifier of an AuxPlayer. Default value: 0

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

AuxPlayerName

Name of the AuxPlayer.

Basis datatype	MaxSize
String	30

2.1.34 AuxPlayerStatus (0x4AC)

Section Type: Extension

This property shows the actual status of the AuxPlayer, selected by function AuxPlayer (0x4AA).

2.1.34.1 Format of Function

Function classes: Sequence Property

FBlock	Function	OPType	Parameter
AuxIn (0x24)	AuxPlayerStatus (0x4AC)	Get	-
		Status	AuxPlayerID , AuxPlayerStatus
		Error	ErrorCode, ErrorInfo

2.1.34.2 Parameter

AuxPlayerID

Identifier of an AuxPlayer. Default value: 0

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	0..65535	1	none

AuxPlayerStatus

Status of the AuxPlayer.

Basis datatype	Range of values	Code	Description
Enum	0x0..0x3	0x0	not active
		0x1	active
		0x2	auto paused
		0x3	busy

2.1.35 DeckStatusSet (0x4D0)

Section Type: Extension

This method replaces the Set OPType of the property DeckStatus (0x200).

2.1.35.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	DeckStatusSet (0x4D0)	StartResult Ack	SenderHandle , DeckStatus
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCodes, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , DeckStatus

2.1.35.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	-		-	none

DeckStatus

- 0x00..0x1F = general states
- 0x20..0x2F = video specific states
- 0x30..0x3F = tape specific states
- 0x40..0x4F = file handling
- 0x50..0x5F = recording

Basis datatype	Range of values	Code	Description
Enum	0x00..0x50	0x00	Play
		0x01	Stop
		0x02	Pause
		0x03	Load / Connect
		0x04	Unload / Disconnect
		0x05	Search Forward
		0x06	Search Backward
		0x07	Fast Forward
		0x08	Fast Backward
		0x09	Empty / No titles available
		0x0A	Retract
		0x20	Slow Motion Forward
		0x21	Slow Motion Backward
		0x22	StepbyStep
		0x23	PreStop
		0x30	Rewind to Begin of Tape
0x31	Forward to End of Tape		
0x32	Search Startposition next title		
0x33	Search Startposition last Title		

Basis datatype	Range of values	Code	Description
		0x40	Fileplay
		0x41	Filetransfer
		0x50	Record

2.1.36 MediaPositionSet (0x4D1)

Section Type: Extension

This method replaces the Set OPType of the property MediaPosition (0x204).

2.1.36.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	MediaPositionSet (0x4D1)	StartResult Ack	SenderHandle , Tag
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCodes, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , Tag

2.1.36.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	-		-	none

Tag

Tag is a unique identifier within the media library.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	none

2.1.37 TitlePositionSet (0x4D2)

Section Type: Extension

This method replaces the Set OPType of the property TitlePosition (0x205).

2.1.37.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	TitlePositionSet (0x4D2)	StartResult Ack	SenderHandle , TitlePosition
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCodes, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , TitlePosition

2.1.37.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	-		-	none

TitlePosition

Number of steps for adjustment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

2.1.38 ChapterPositionSet (0x4D3)

Section Type: Extension

This method replaces the Set OPType of the property ChapterPosition (0x206).

2.1.38.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	ChapterPositionSet (0x4D3)	StartResult Ack	SenderHandle , ChapterPosition
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCodes, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , ChapterPosition

2.1.38.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	-		-	none

ChapterPosition

Number of steps for adjustment.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

2.1.39 TrackInformationSet (0x4D4)

Section Type: Extension

This method replaces the Set OPType of the property TrackInformation (0x434).

2.1.39.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	TrackInformation Set (0x4D4)	StartResult Ack	CurrentRelativeTrackPosition
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCode, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , CurrentNumberTracks , CurrentRelativeTrackPosition

2.1.39.2 Parameter

CurrentRelativeTrackPosition

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	none

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	-		-	none

CurrentNumberTracks

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	none

2.1.40 AuxTrackPositionSet (0x4D5)

Section Type: Extension

This method replaces the Set OPType of the property AuxTrackPosition (0x439).

2.1.40.1 Format of Function

Function classes: Unclassified Method

FBLOCK	Function	OPType	Parameter
AuxIn (0x24)	AuxTrackPosition Set (0x4D5)	StartResult Ack	SenderHandle , Track
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCodes, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , Track

2.1.40.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	-		-	none

Track

The current track.

Note:

- Track=0 for "no track", e.g. if there is no medium available.
- For a tape player, Track=1 corresponds to the first side of the tape and Track=2 corresponds to the second side.
- The currently valid range of values is dependent on the medium. When required, the actual range is queriable by the interface.
- 0xFFFF FFFF: exact track number not known (e.g., AuxIn device is still calculating track number)

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	none

2.1.41 RandomSet (0x4D6)

Section Type: Extension

This method replaces the Set OPType of the property Random (0x450).

2.1.41.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	RandomSet (0x4D6)	StartResult Ack	SenderHandle , RandomState
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCodes, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , RandomState

2.1.41.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	-		-	none

RandomState

Setting the RandomState to "All" can cause a implicit change of the used filter for playback, if the filter was not set to all media on the connected device. The changed filter can be signaled by the gateway using notification for CurrentAudioListFilter property.

Basis datatype	Range of values	Code	Description
Enum	0x00..0x0C	0x00	Off
		0x01	reserved
		0x02	Disk
		0x03	Magazine / Albums
		0x04	All / All Magazines
		0x05 ... 0x09	Reserved
		0x0A	Folder
		0x0B	SubFolder
		0x0C	Tracklist

2.1.42 ScanSet (0x4D7)

Section Type: Extension

This method replaces the Set OPType of the property Scan (0x451).

2.1.42.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	ScanSet (0x4D7)	StartResult Ack	SenderHandle , ScanState
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCodes, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , ScanState

2.1.42.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	-		-	none

ScanState

Basis datatype	Range of values	Code	Description
Enum	0x00..0x0C	0x00	Off
		0x01	reserved
		0x02	Disk
		0x03	Magazine
		0x04	All / All Magazines
		0x05 ... 0x09	Reserved
		0x0A	Folder
		0x0B	SubFolder
		0x0C	Tracklist

2.1.43 RepeatSet (0x4D8)

Section Type: Extension

This method replaces the Set OPType of the property Repeat (0x452).

2.1.43.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
AuxIn (0x24)	RepeatSet (0x4D8)	StartResult Ack	SenderHandle , RepeatState
		AbortAck	SenderHandle
		ErrorAck	SenderHandle , ErrorCodes, ErrorInfo
		Processing Ack	SenderHandle
		ResultAck	SenderHandle , RepeatState

2.1.43.2 Parameter

SenderHandle

Unique handle to identify the request.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	-		-	none

RepeatState

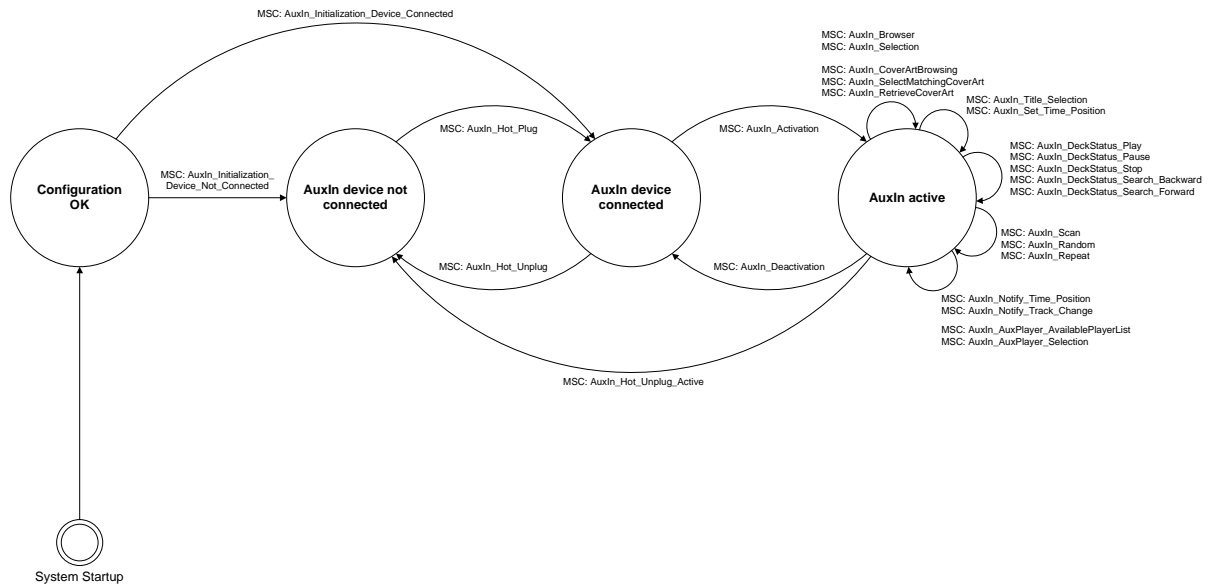
Setting the RepeatState to "All" can cause an implicit change of the used filter for playback, if the filter was not set to all media on the connected device. The changed filter can be signaled by the gateway using notification for CurrentAudioListFilter property.

Basis datatype	Range of values	Code	Description
Enum	0x00..0x0C	0x00	Off
		0x01	Track
		0x02	Disk
		0x03	Magazine
		0x04	All / All magazines
		0x05	Chapter
		0x06	Title
		0x07	AB
		0x08 ... 0x09	Reserved
		0x0A	Folder
		0x0B	SubFolder
		0x0C	Tracklist

3 Dynamic Specification

The sequence relation chart (SRC) gives an overview how the sequences (MSCs) are related and depend on each other. The bubbles identify the main and common states of the FBlock together with its controller(s). The arrows are labeled with MSCs and specify the interaction (MSC) for the transitions from one state to the next.

Note: The chart does not specify the complete behavior of the FBlock AuxIn and its controllers. It describes the core use cases.

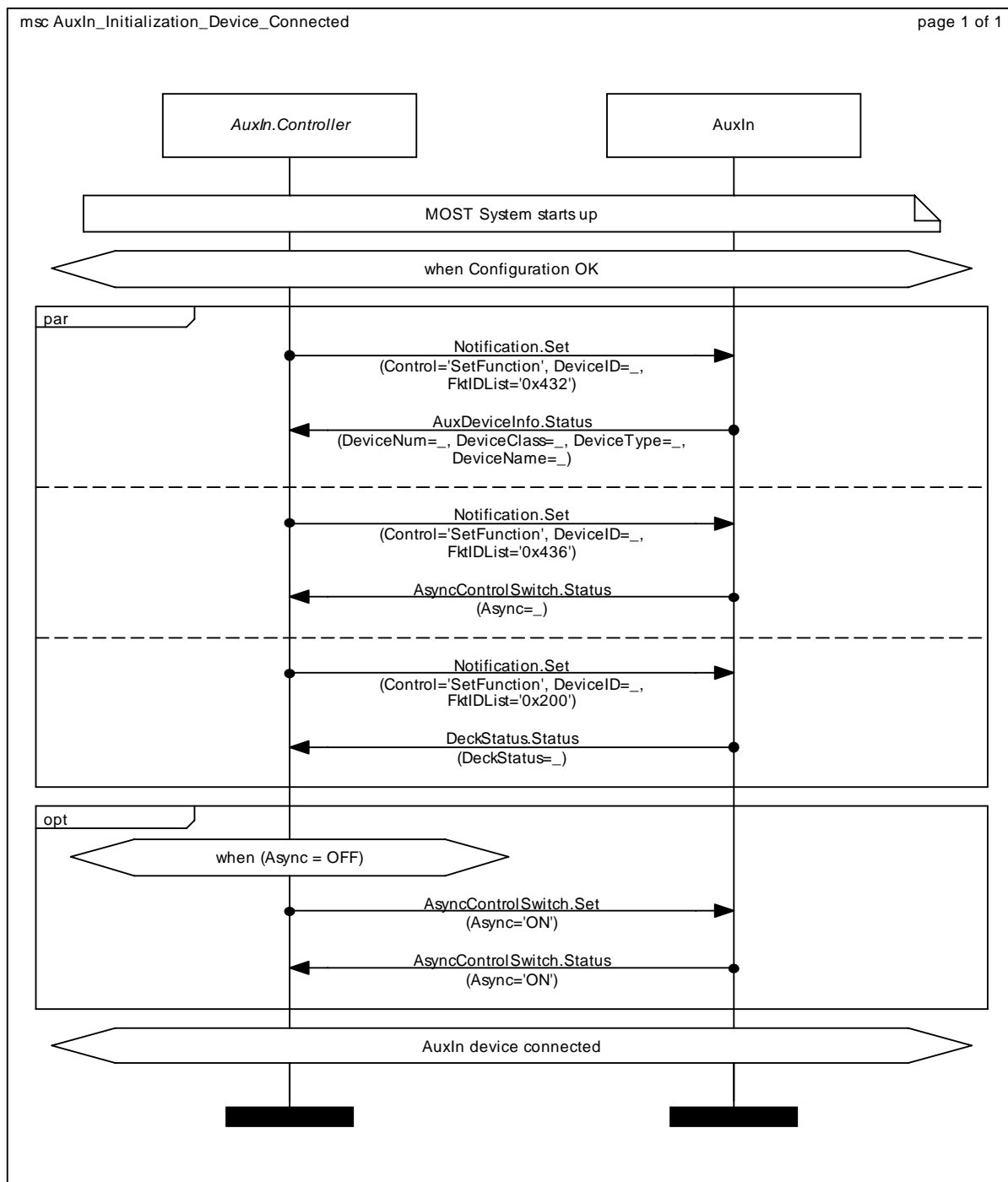


Revision: 1.1 Date: 2008/09/12

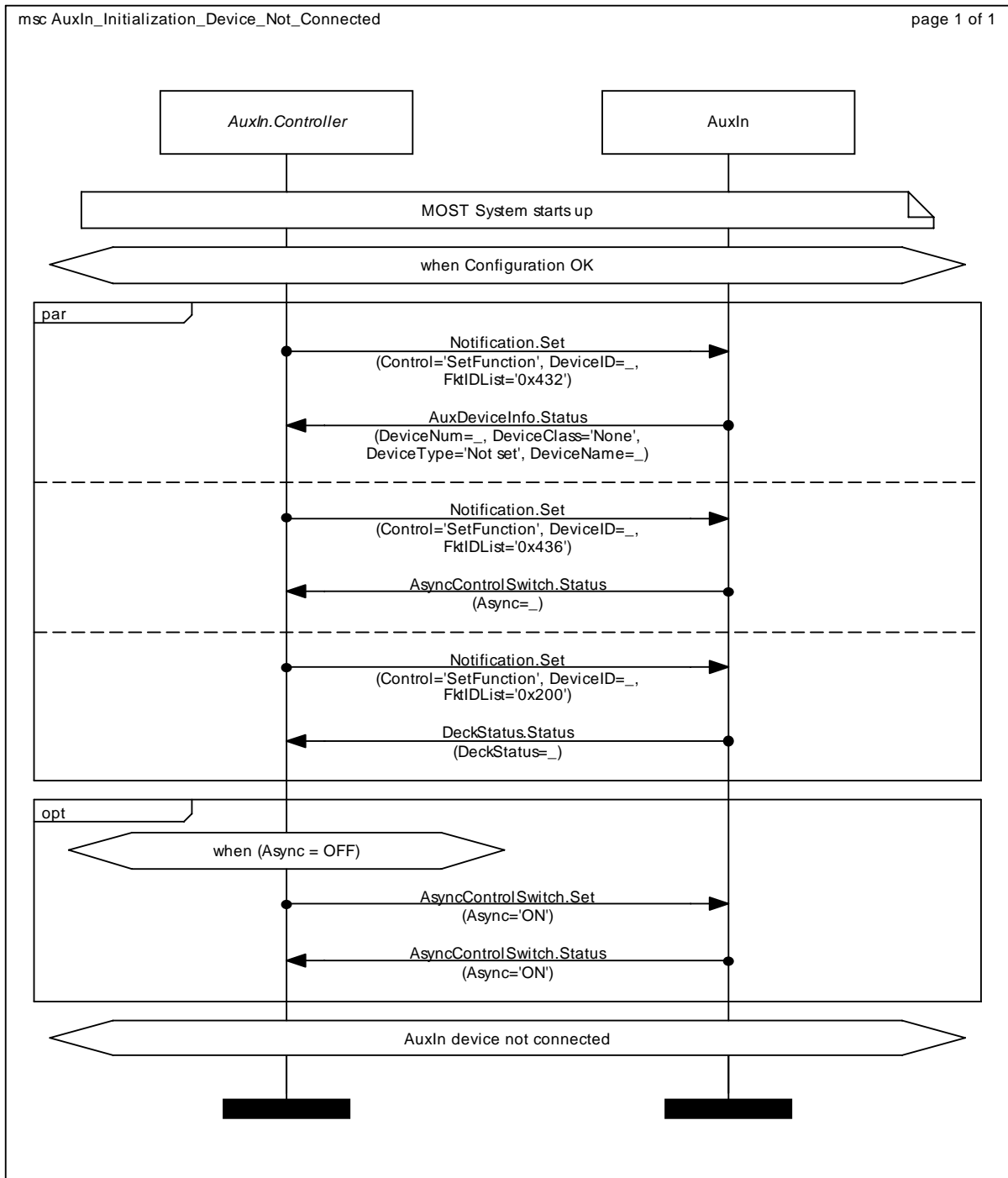
RCSFile: AuxIn_v35_Seq_Relation_Chart.vsd

3.1 AuxIn Startup

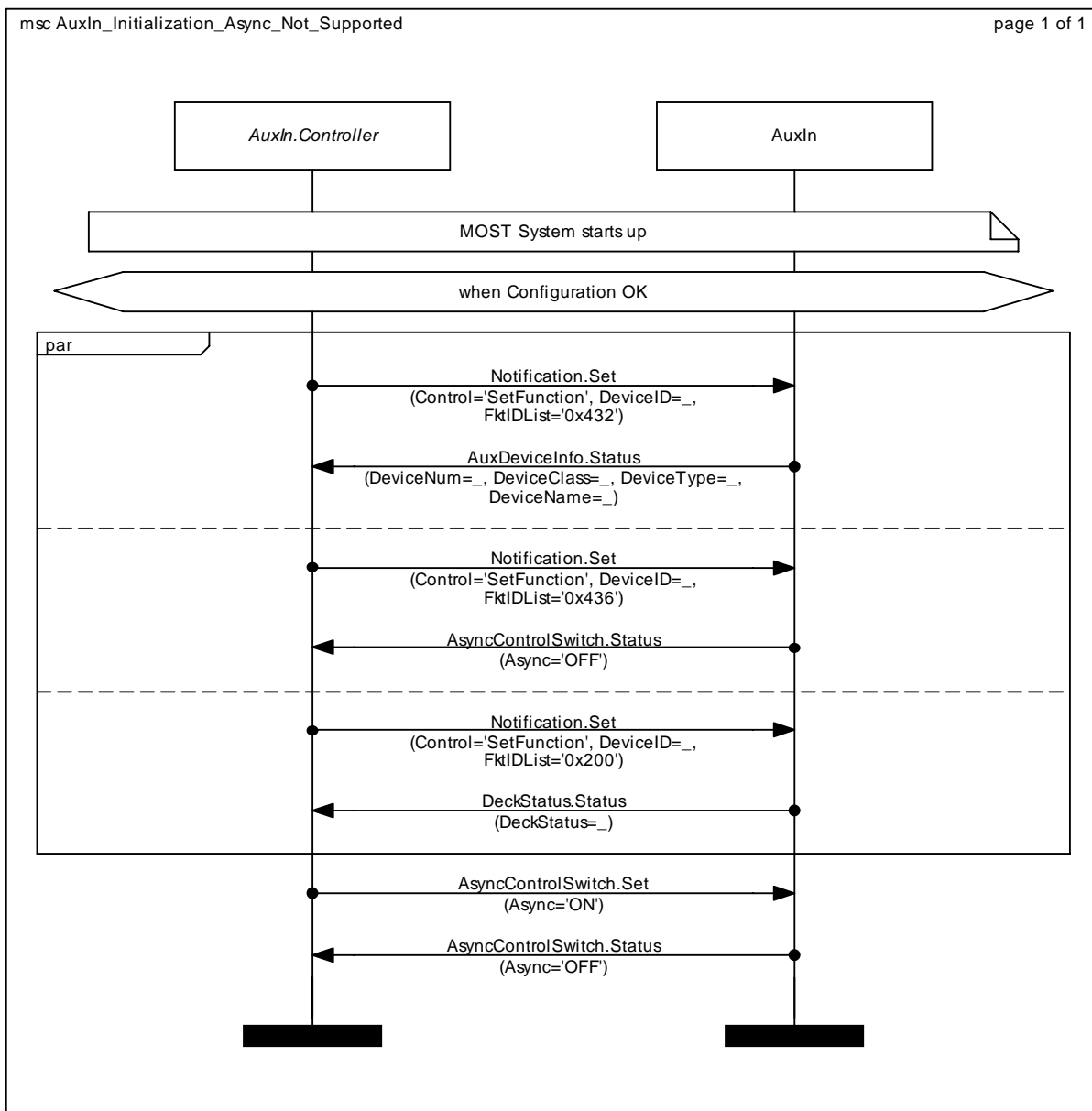
3.1.1 AuxIn_Initialization_Device_Connected



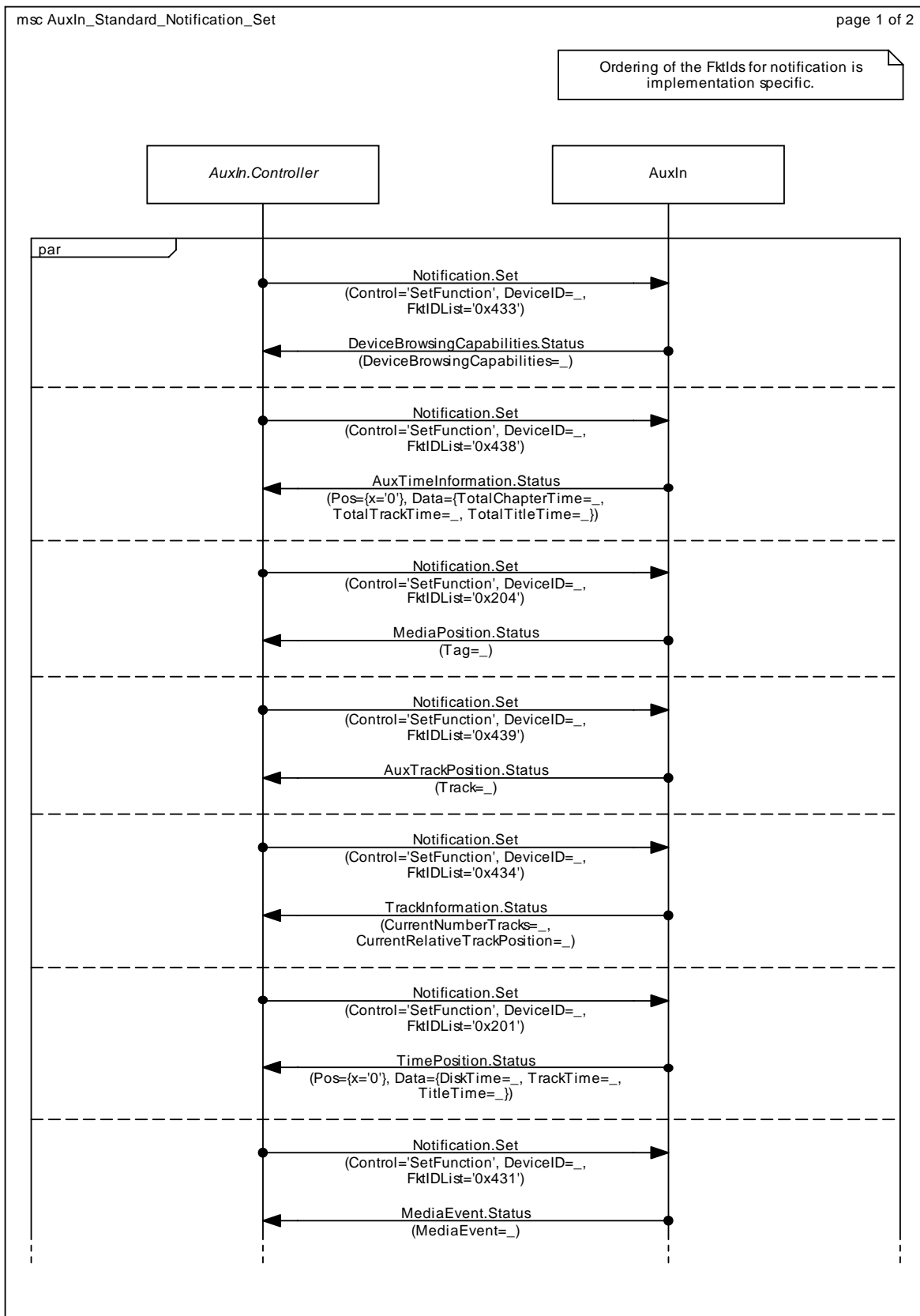
3.1.2 AuxIn_Initialization_Device_Not_Connected

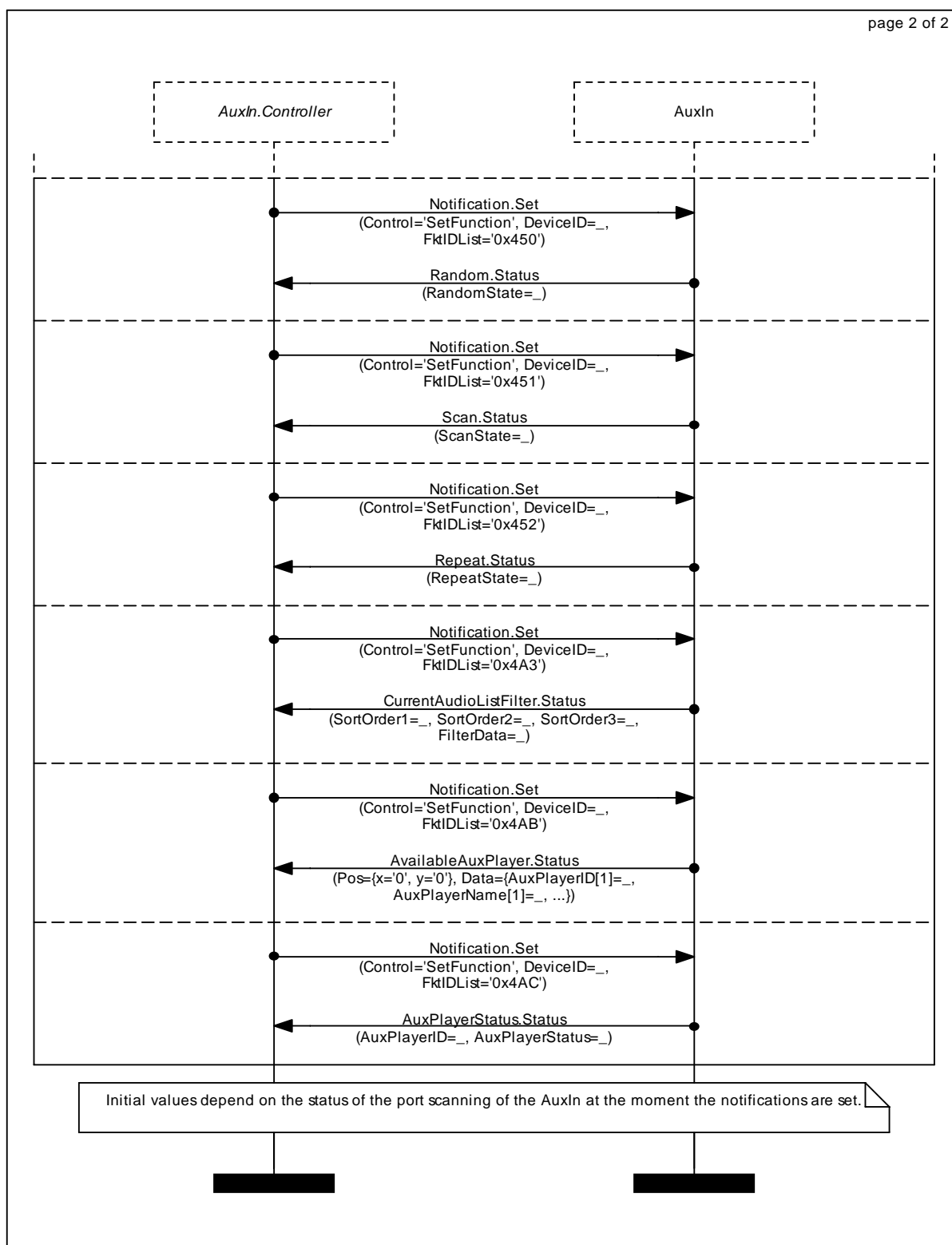


3.1.3 AuxIn_Initialization_Async_Not_Supported

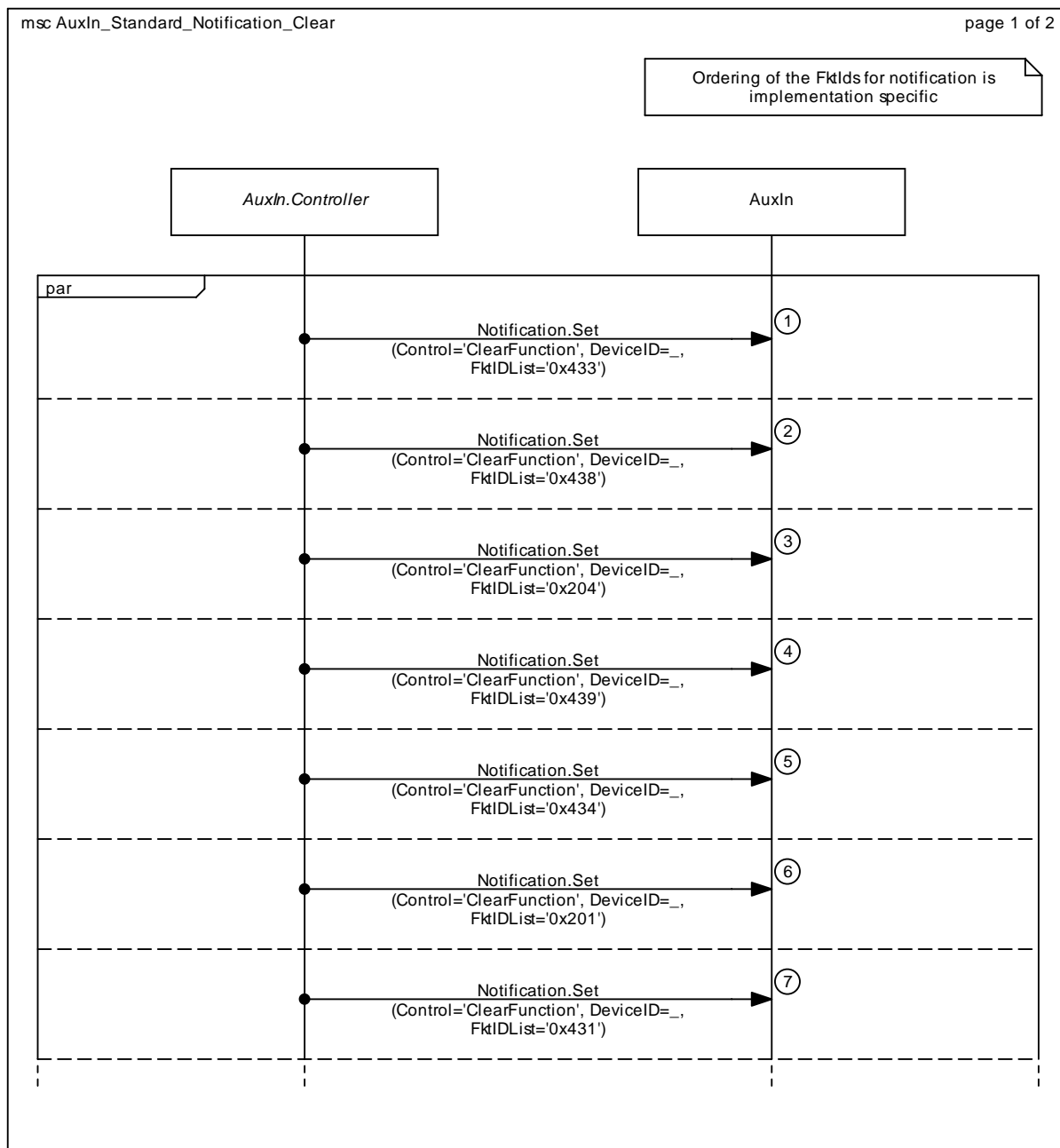


3.1.4 AuxIn_Standard_Notification_Set

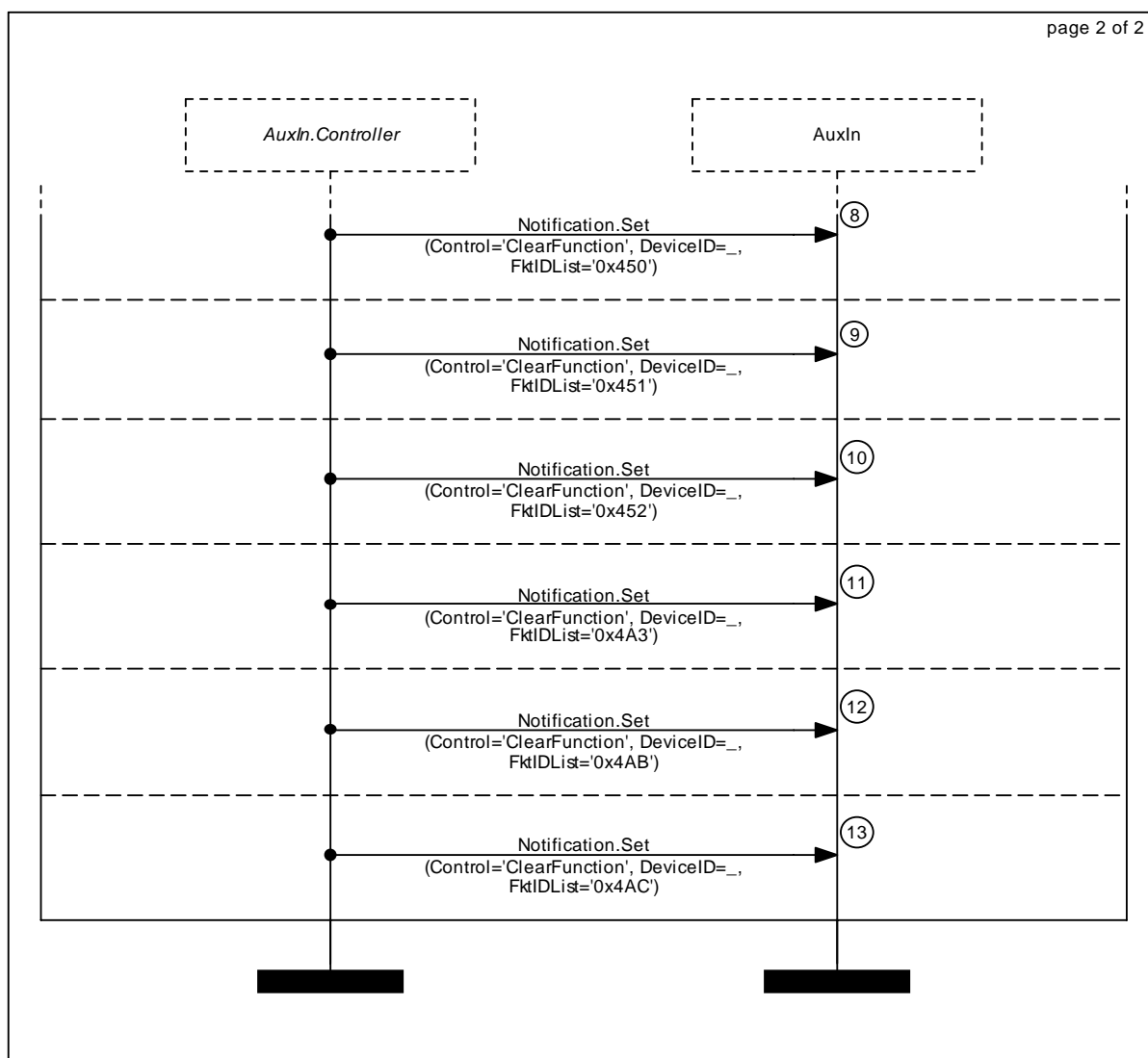




3.1.5 AuxIn_Standard_Notification_Clear



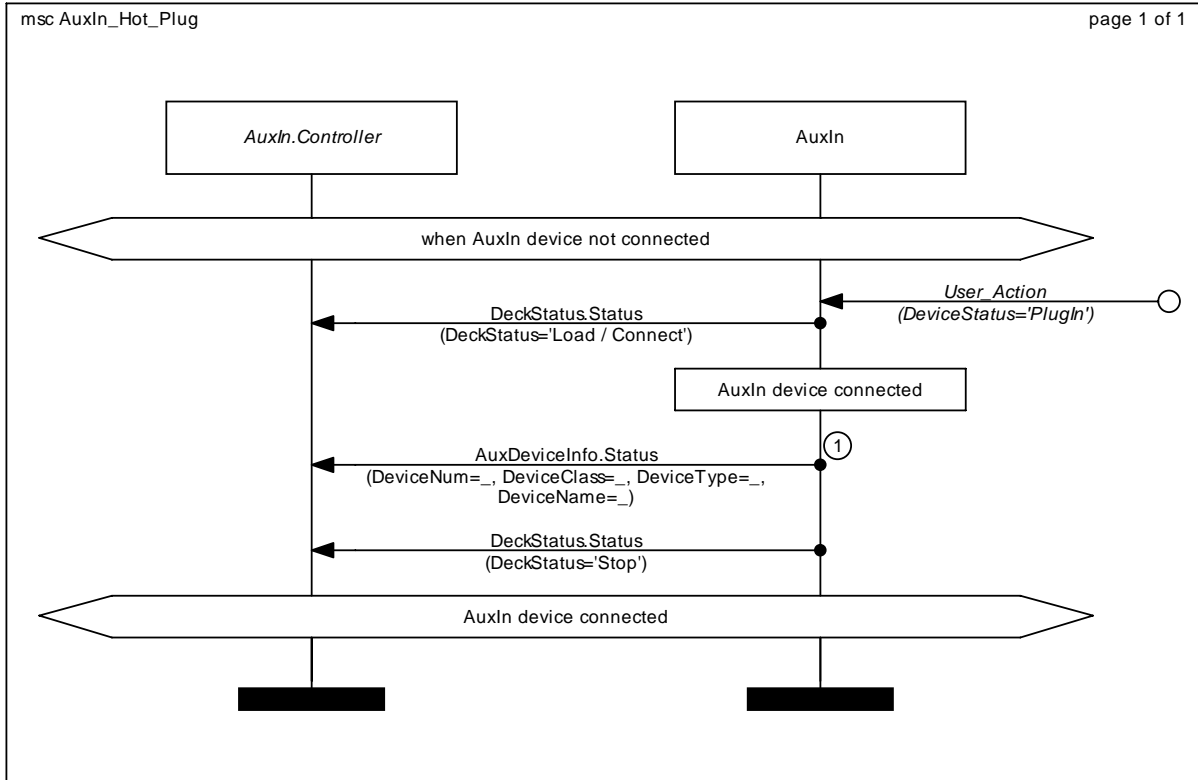
- 1 0x433 - DeviceBrowsingCapabilities
- 2 0x438 - AuxTimeInformation
- 3 0x204 - MediaPosition
- 4 0x439 - AuxTrackPosition
- 5 0x434 - TrackInformation
- 6 0x201 - TimePosition
- 7 0x431 - MediaEvent



- 8 0x450 - Random
- 9 0x451 - Scan
- 10 0x452 - Repeat
- 11 0x4A3 - CurrentAudioListFilter
- 12 0x4AB - AvailableAuxPlayer
- 13 0x4AC - AuxPlayerStatus

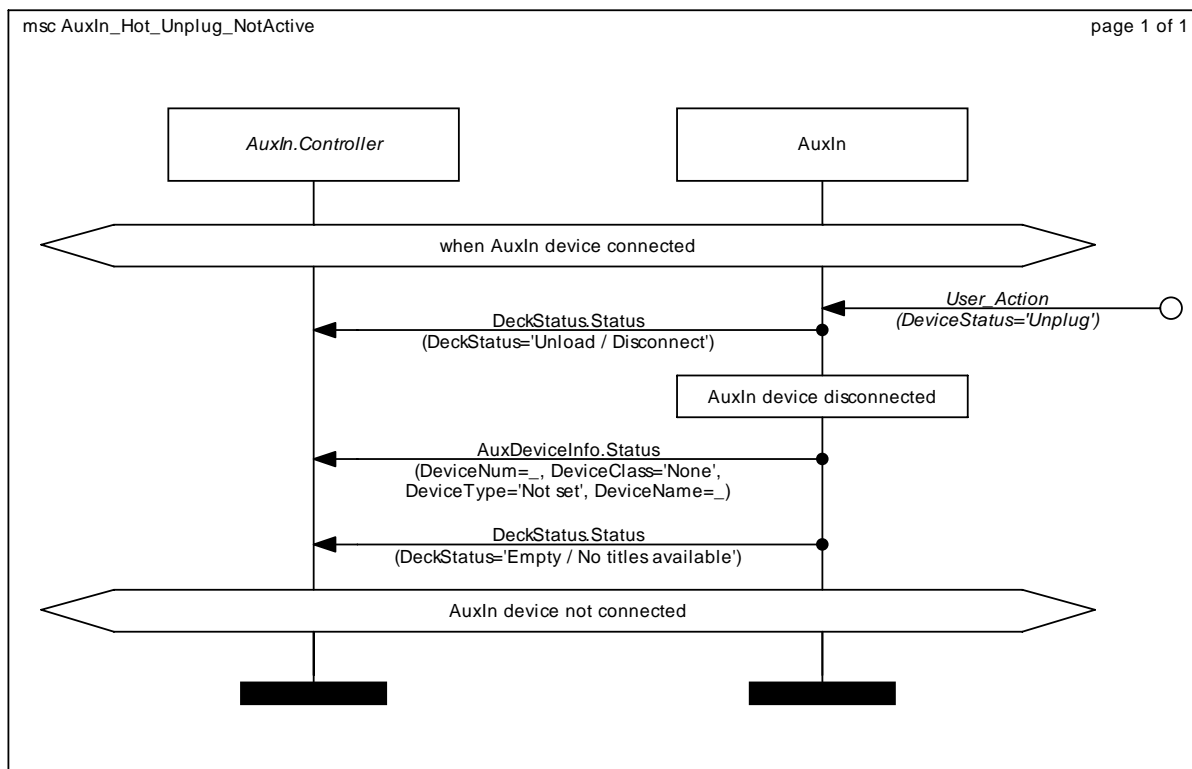
3.2 Connecting an External Device

3.2.1 AuxIn_Hot_Plug

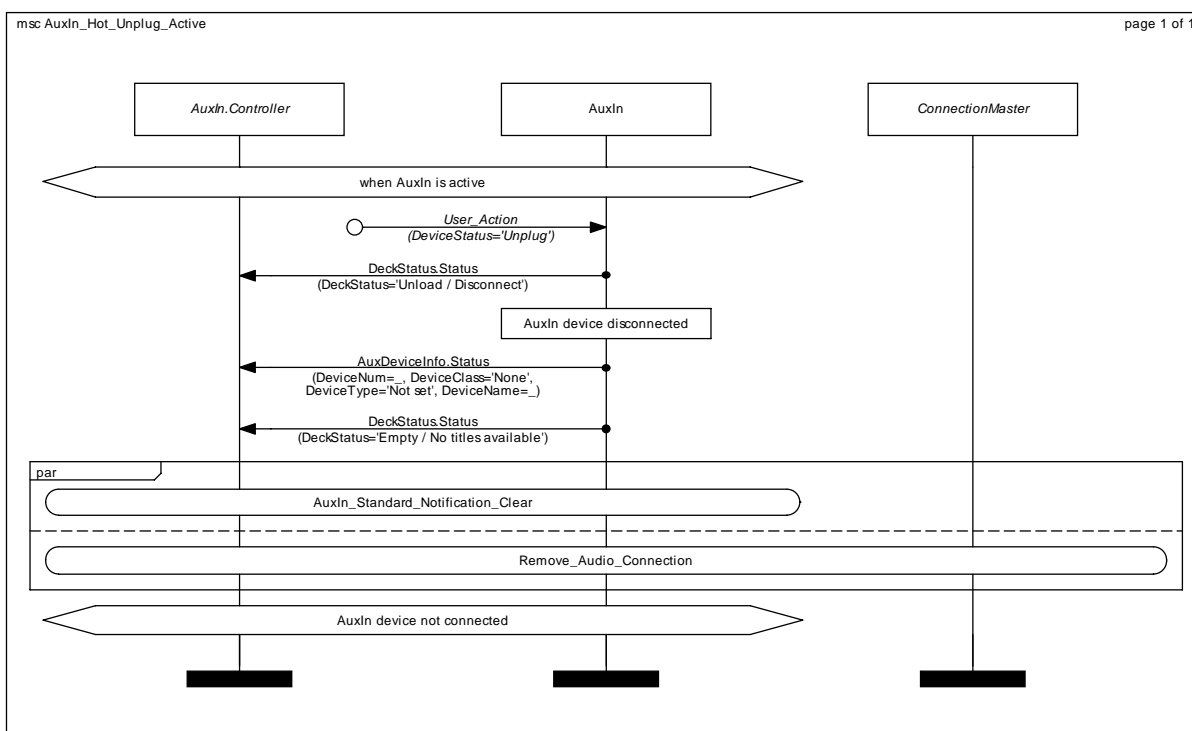


- 1 DeviceClass != 0x00 (None),
DeviceType != 0xFF (Not Set)

3.2.2 AuxIn_Hot_Unplug_NotActive

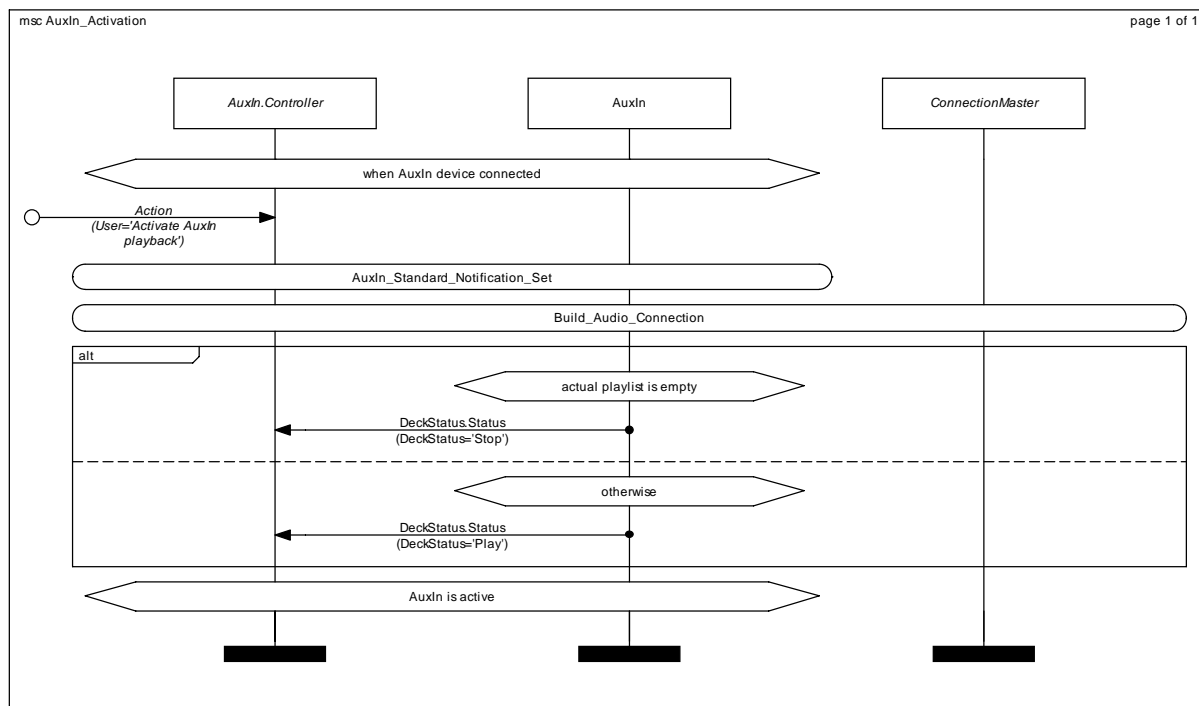


3.2.3 AuxIn_Hot_Unplug_Active

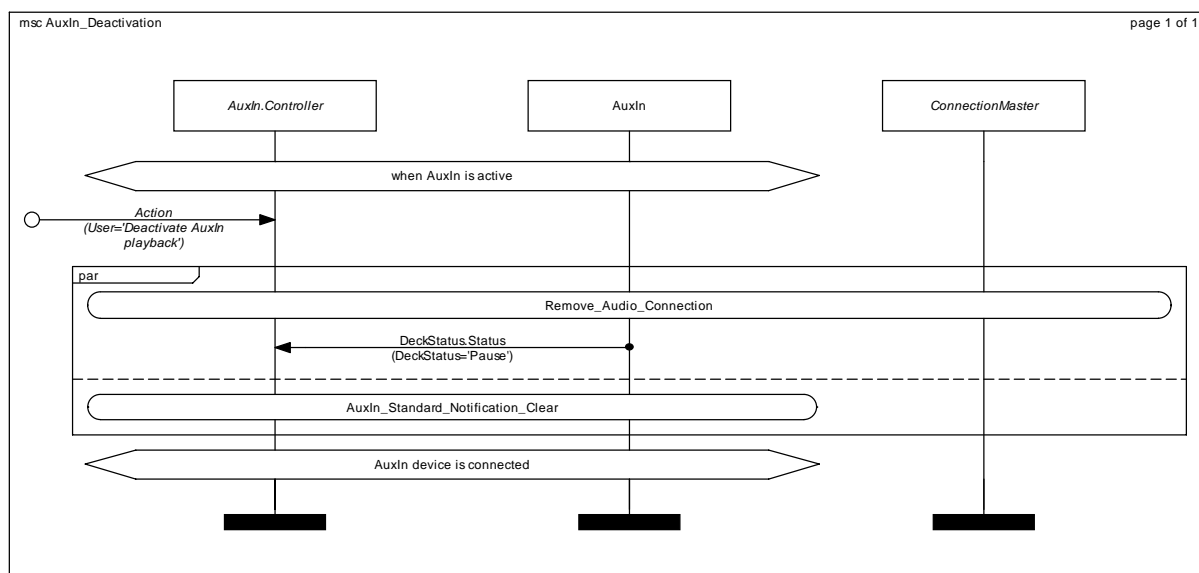


3.3 Audio Source Handling

3.3.1 AuxIn_Activation

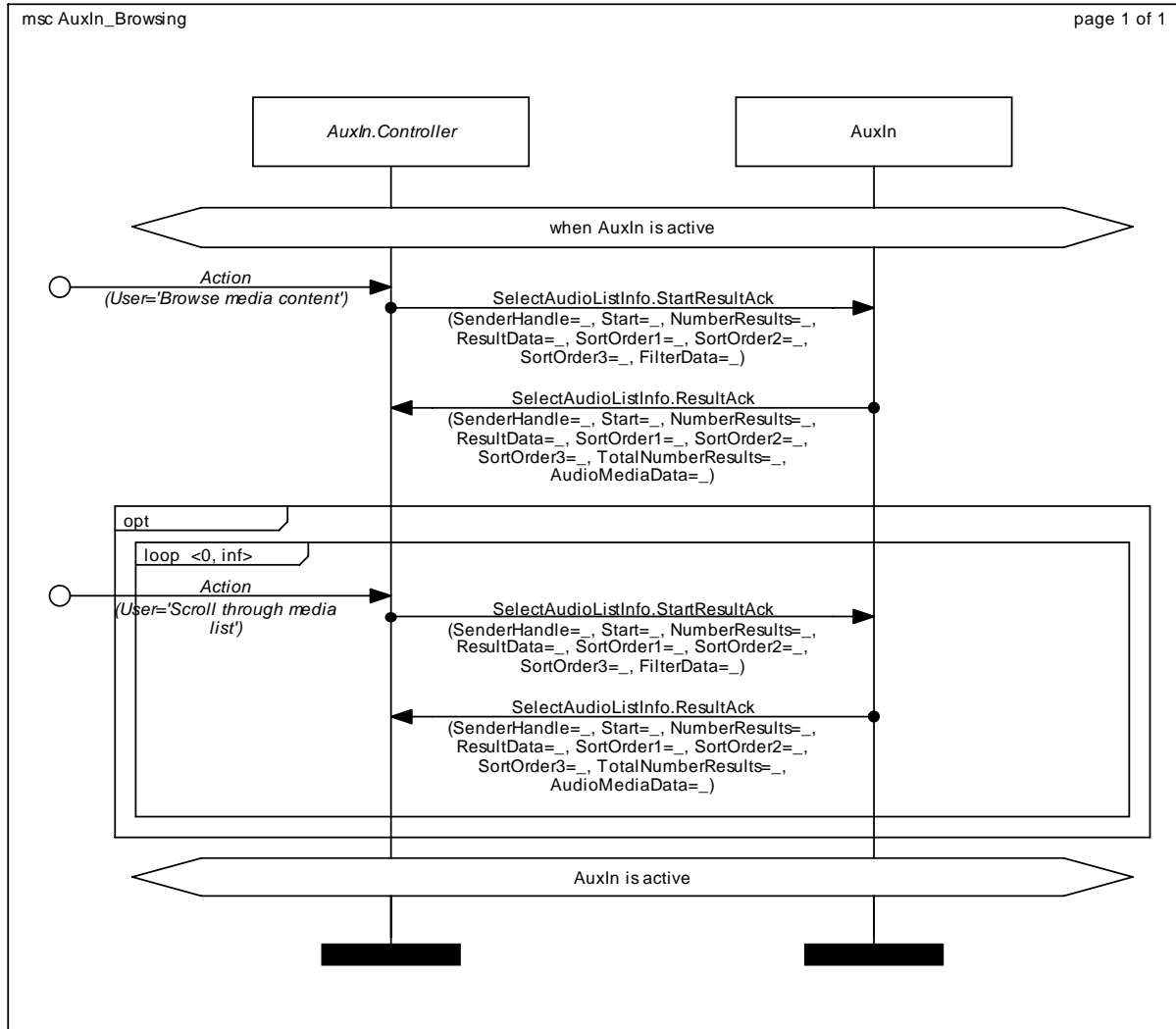


3.3.2 AuxIn_Deactivation

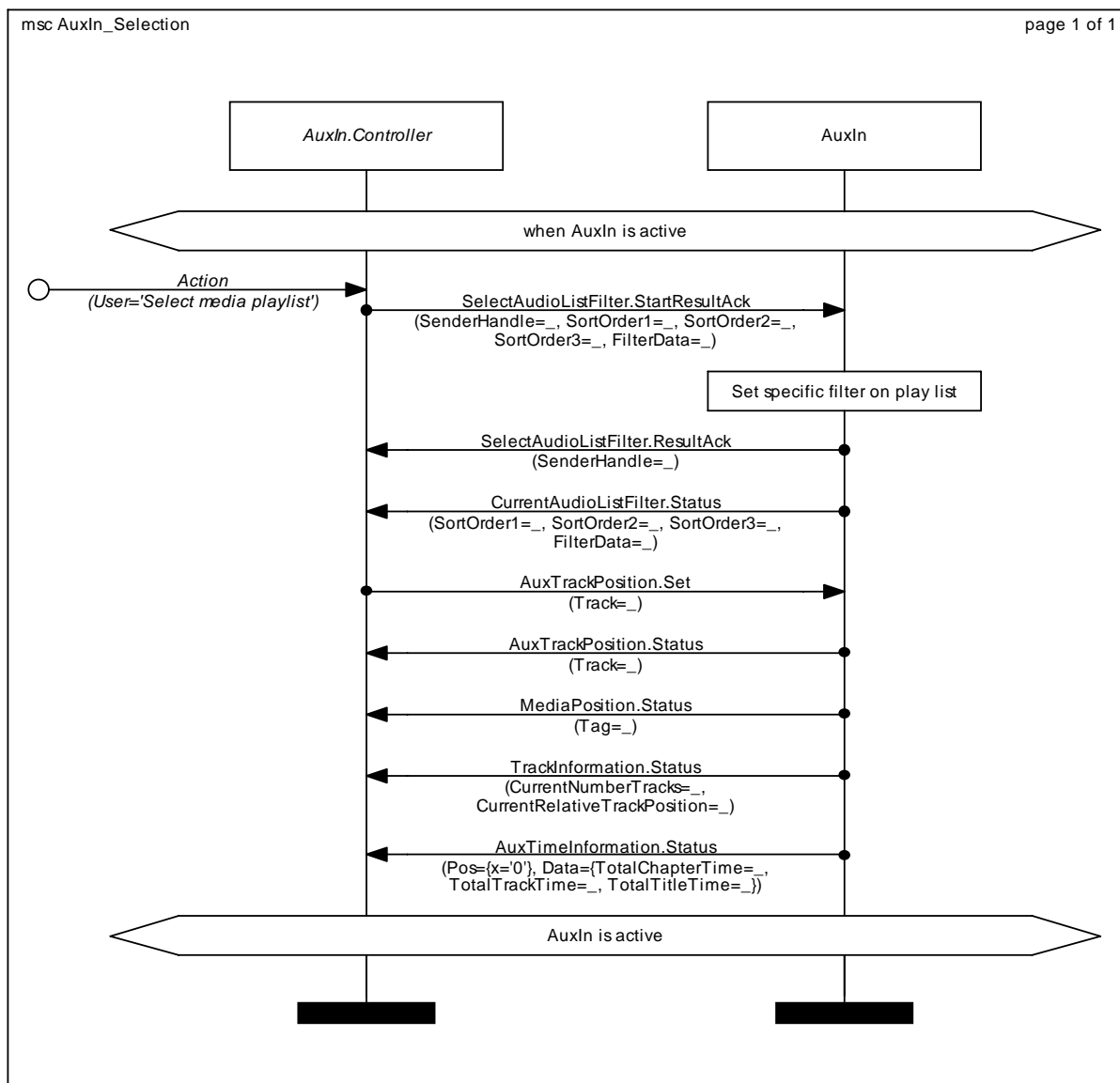


3.4 Managing Media Lists

3.4.1 AuxIn_Browsing

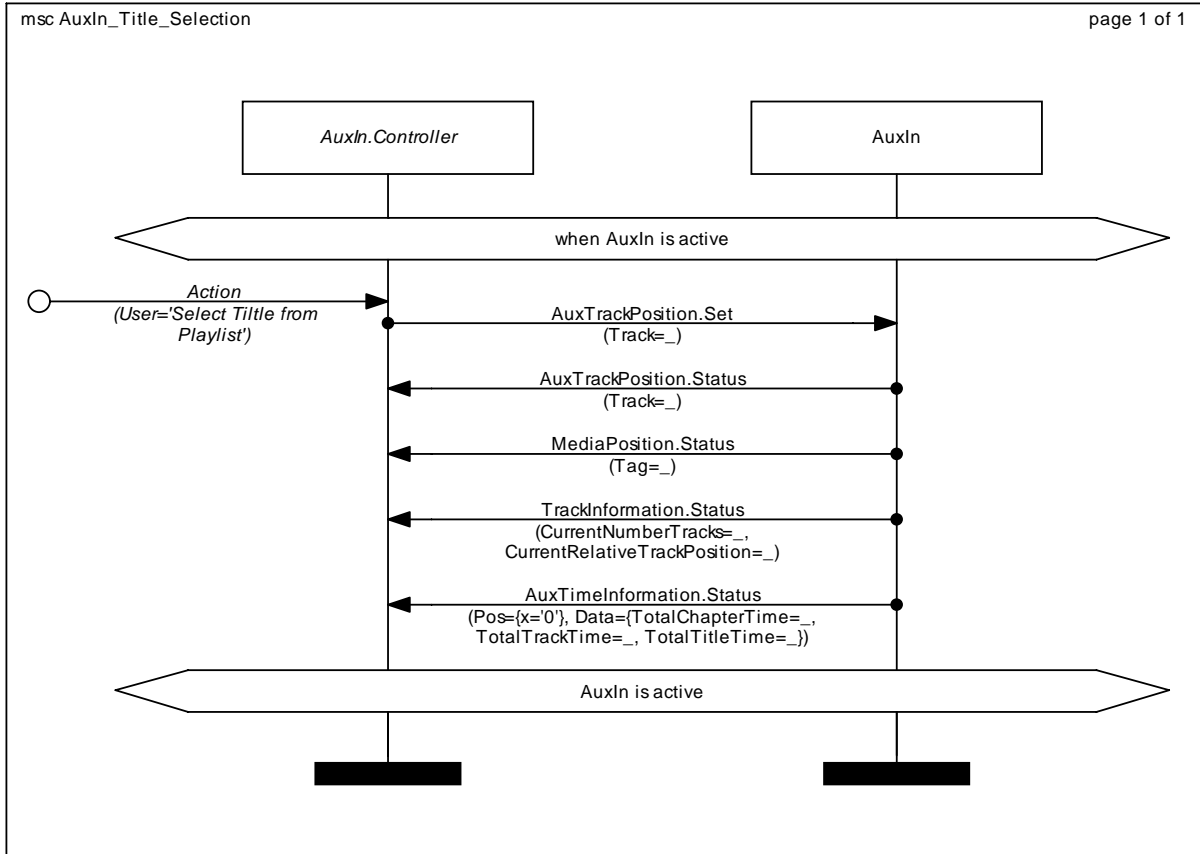


3.4.2 AuxIn_Selection

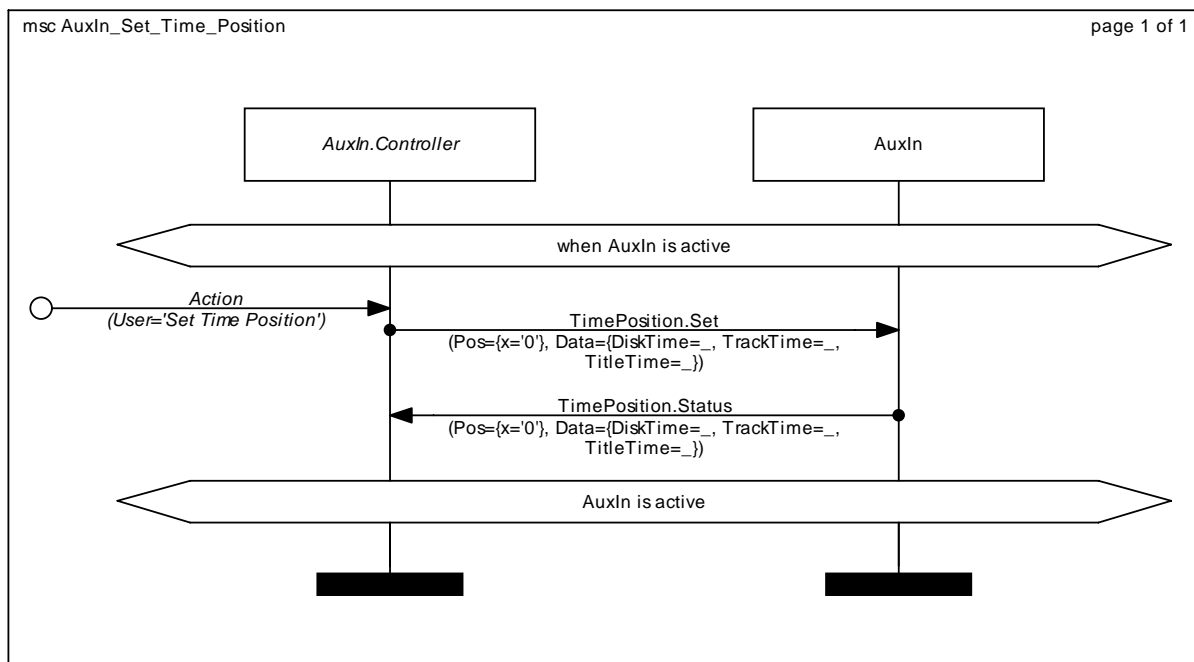


3.5 AuxIn Control

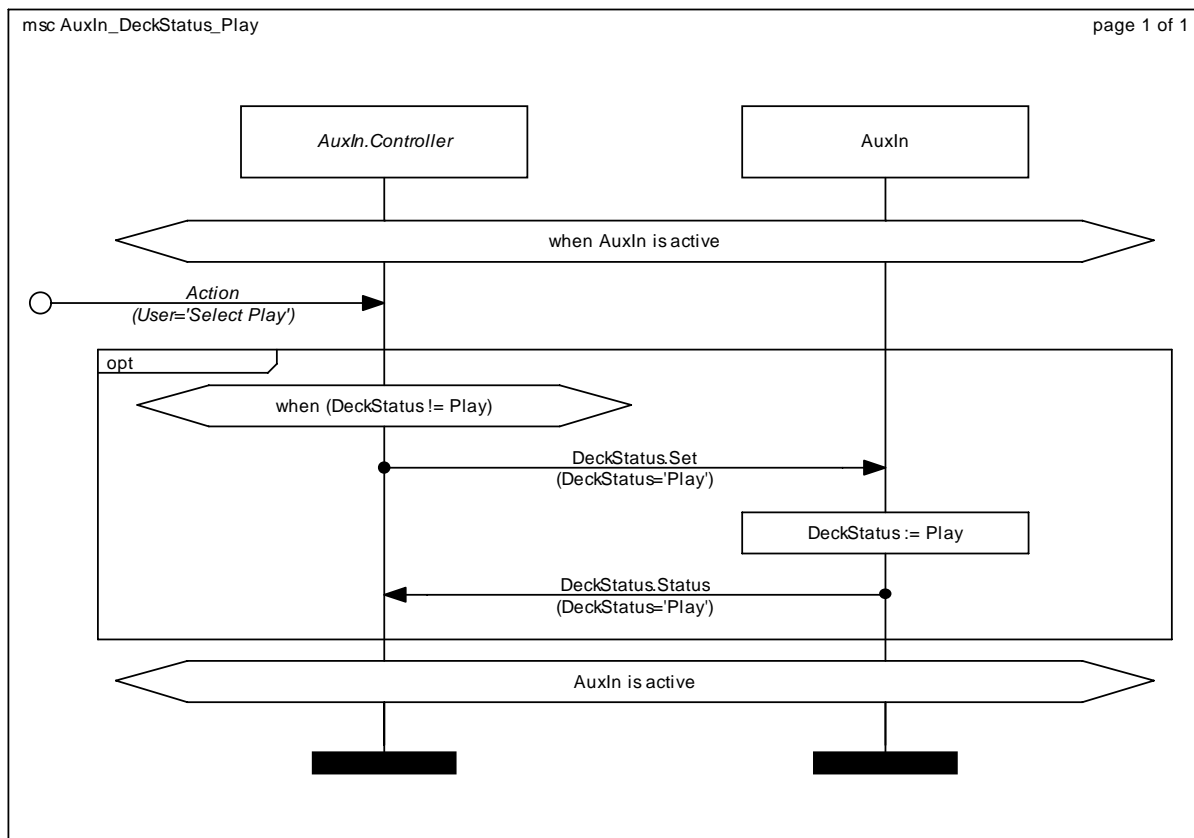
3.5.1 AuxIn_Title_Selection



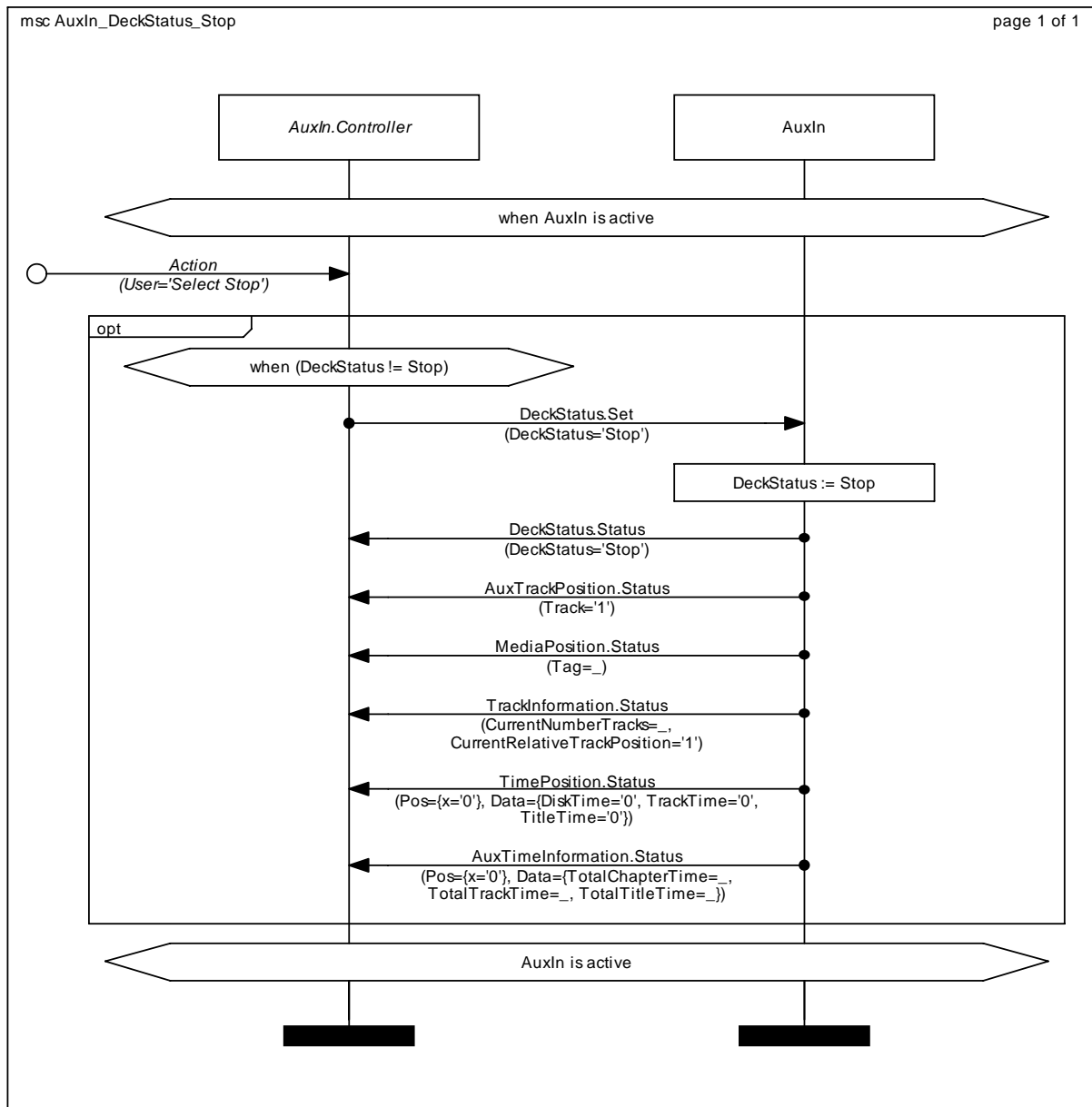
3.5.2 AuxIn_Set_Time_Position



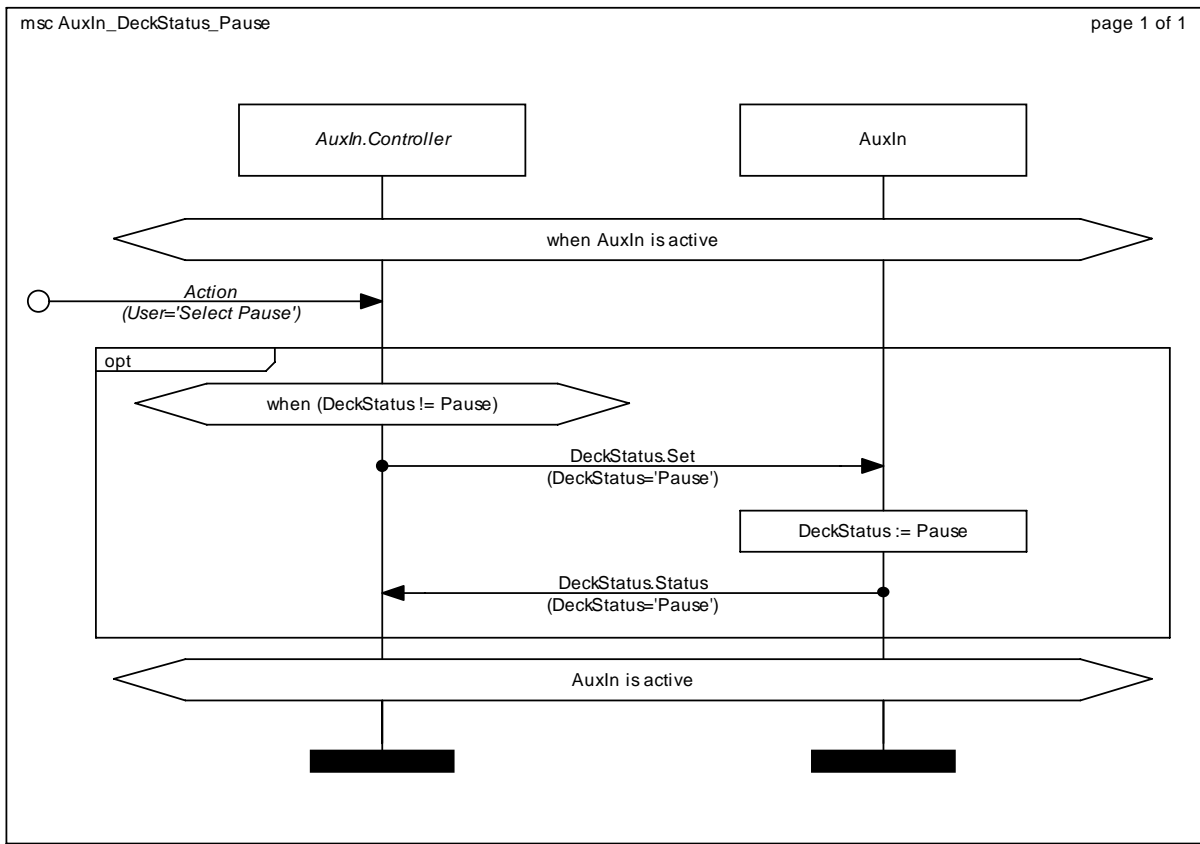
3.5.3 AuxIn_DeckStatus_Play



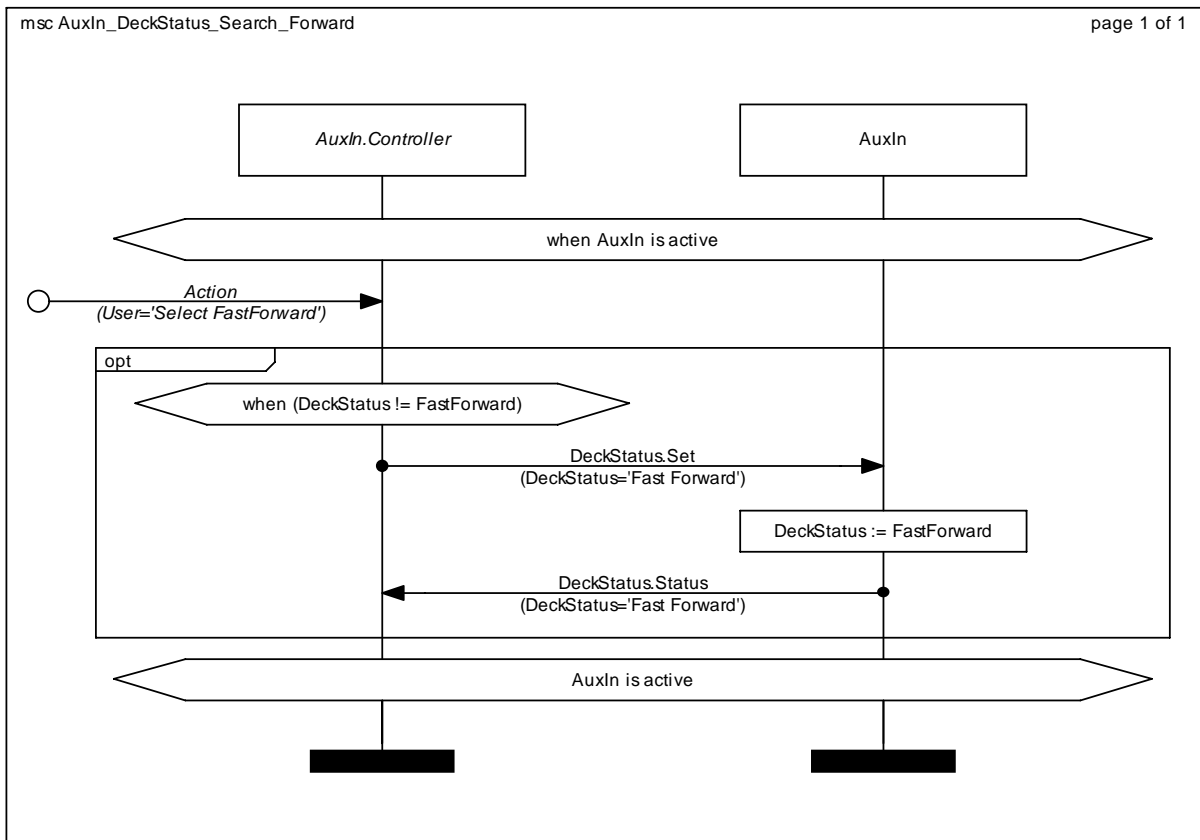
3.5.4 AuxIn_DeckStatus_Stop



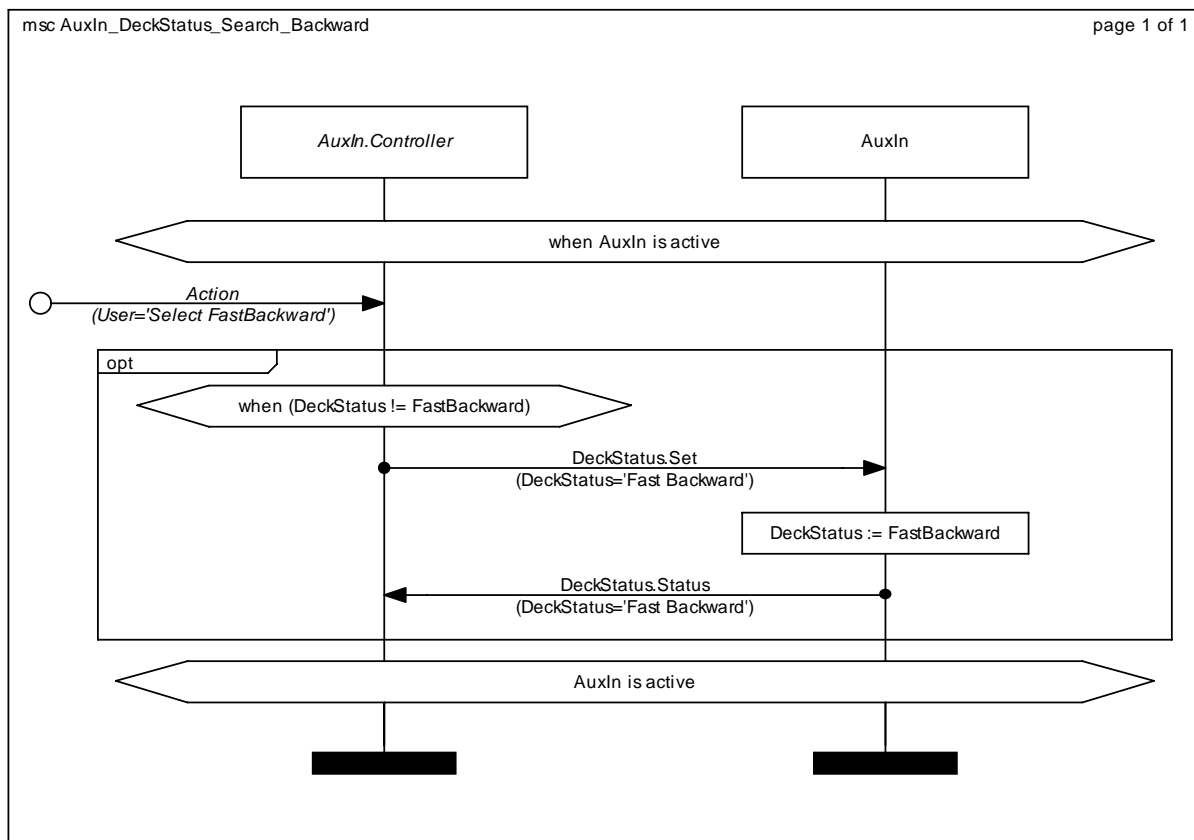
3.5.5 AuxIn_DeckStatus_Pause



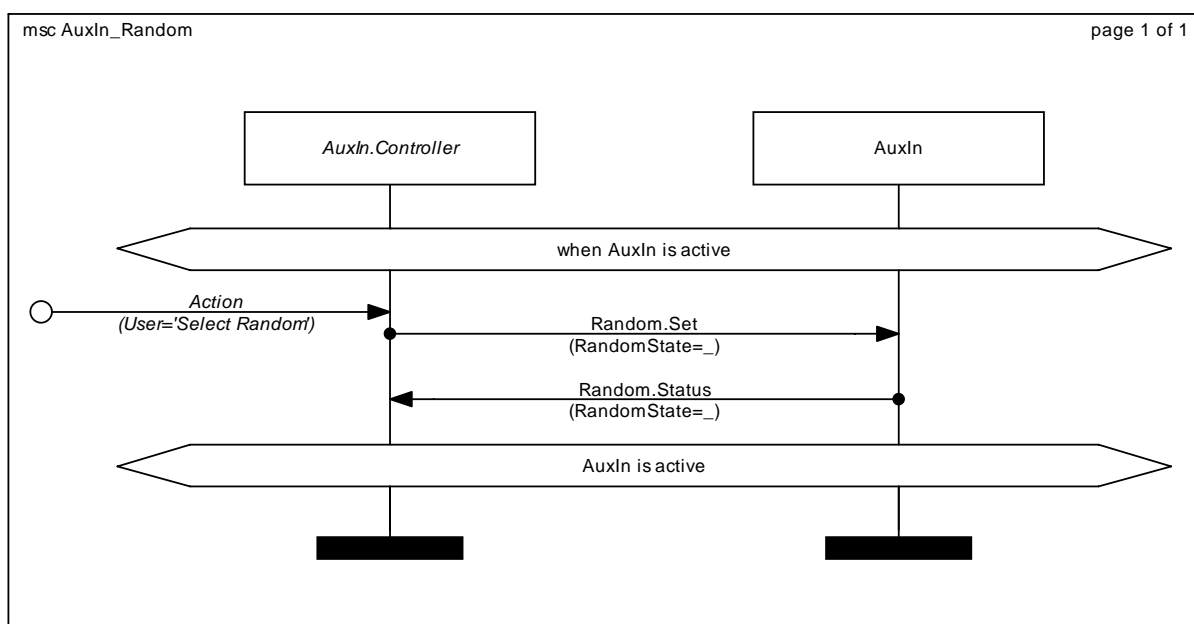
3.5.6 AuxIn_DeckStatus_Search_Forward



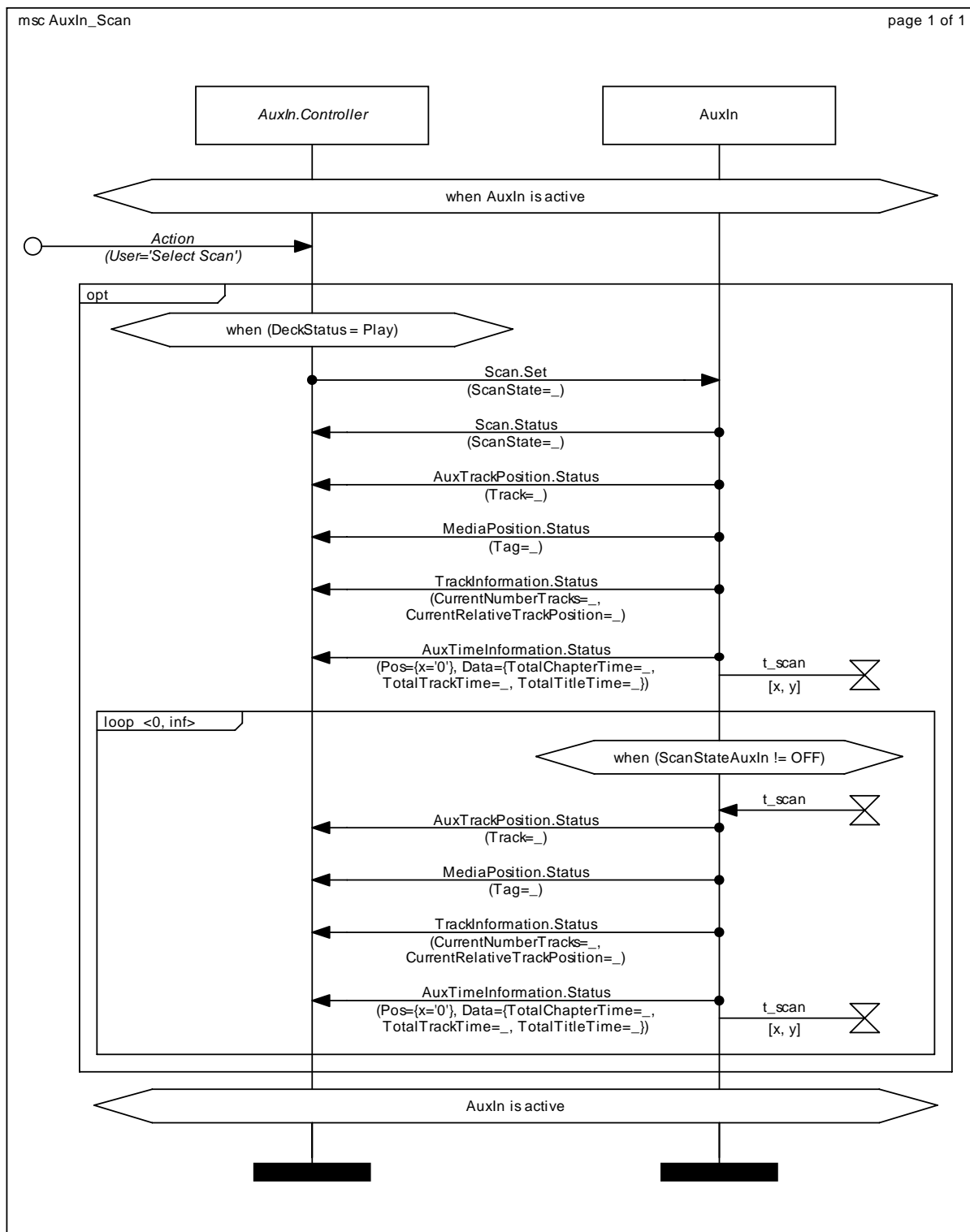
3.5.7 AuxIn_DeckStatus_Search_Backward



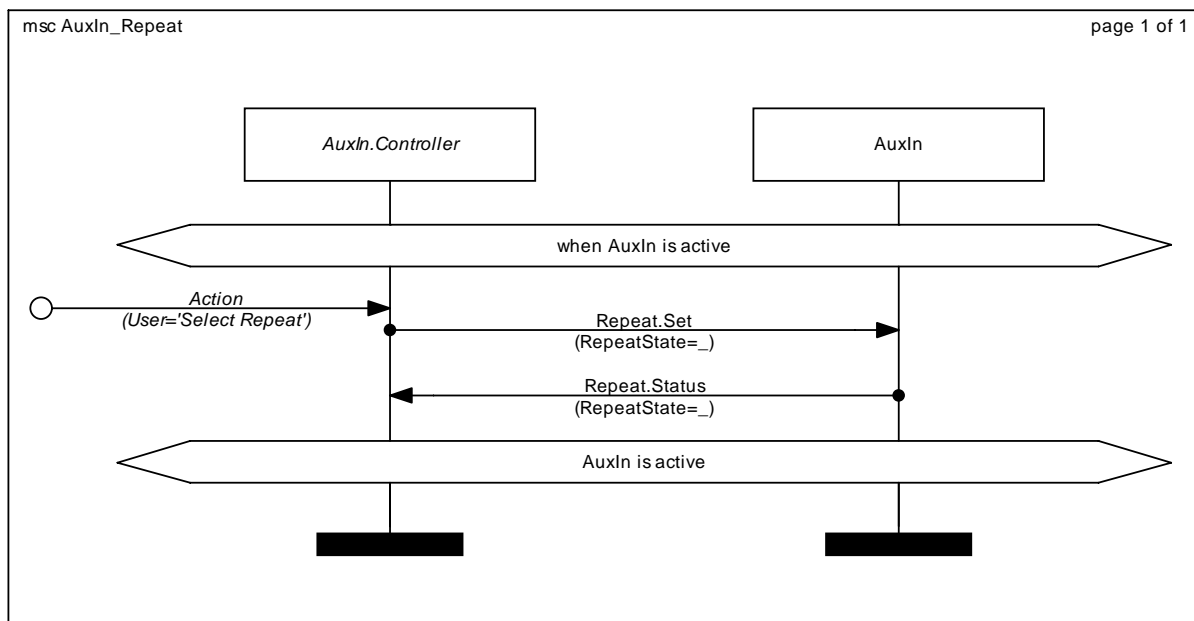
3.5.8 AuxIn_Random



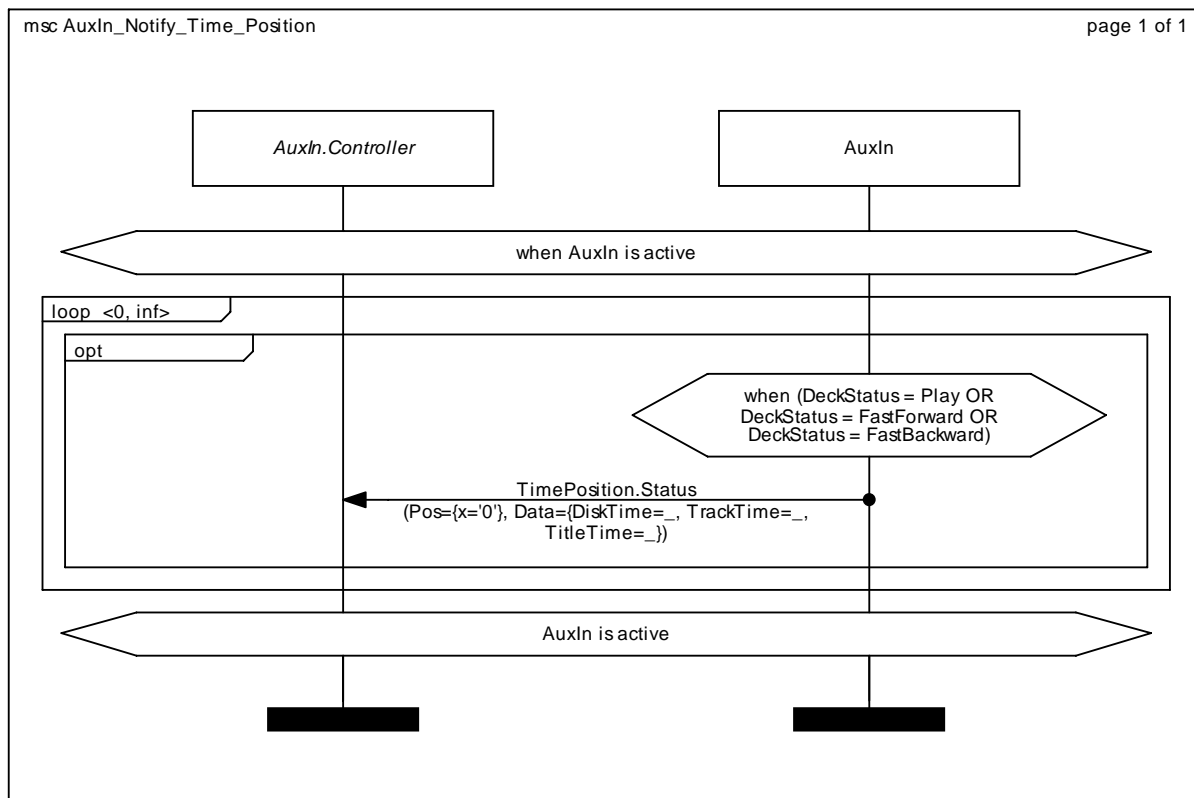
3.5.9 AuxIn_Scan



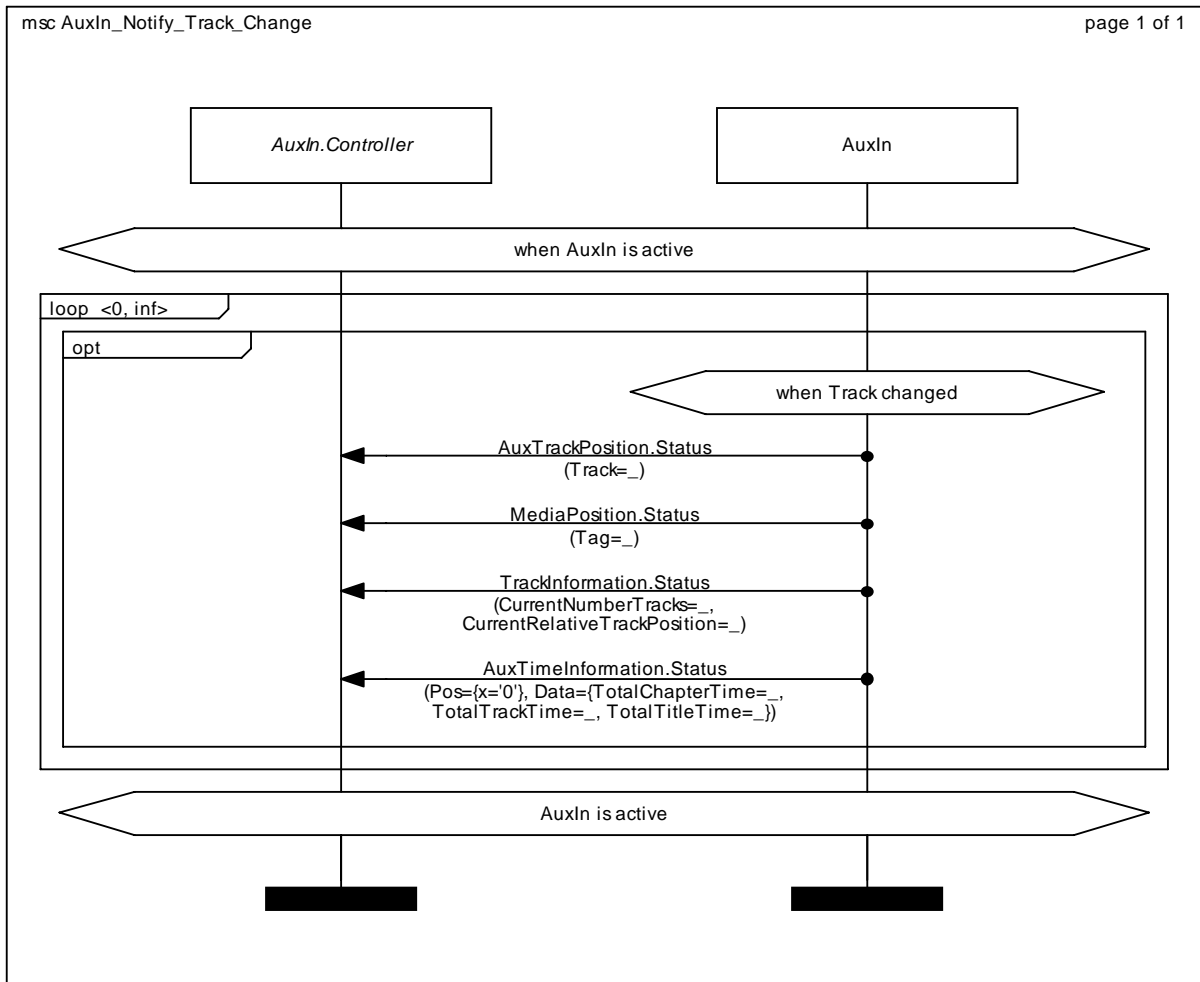
3.5.10 AuxIn_Repeat



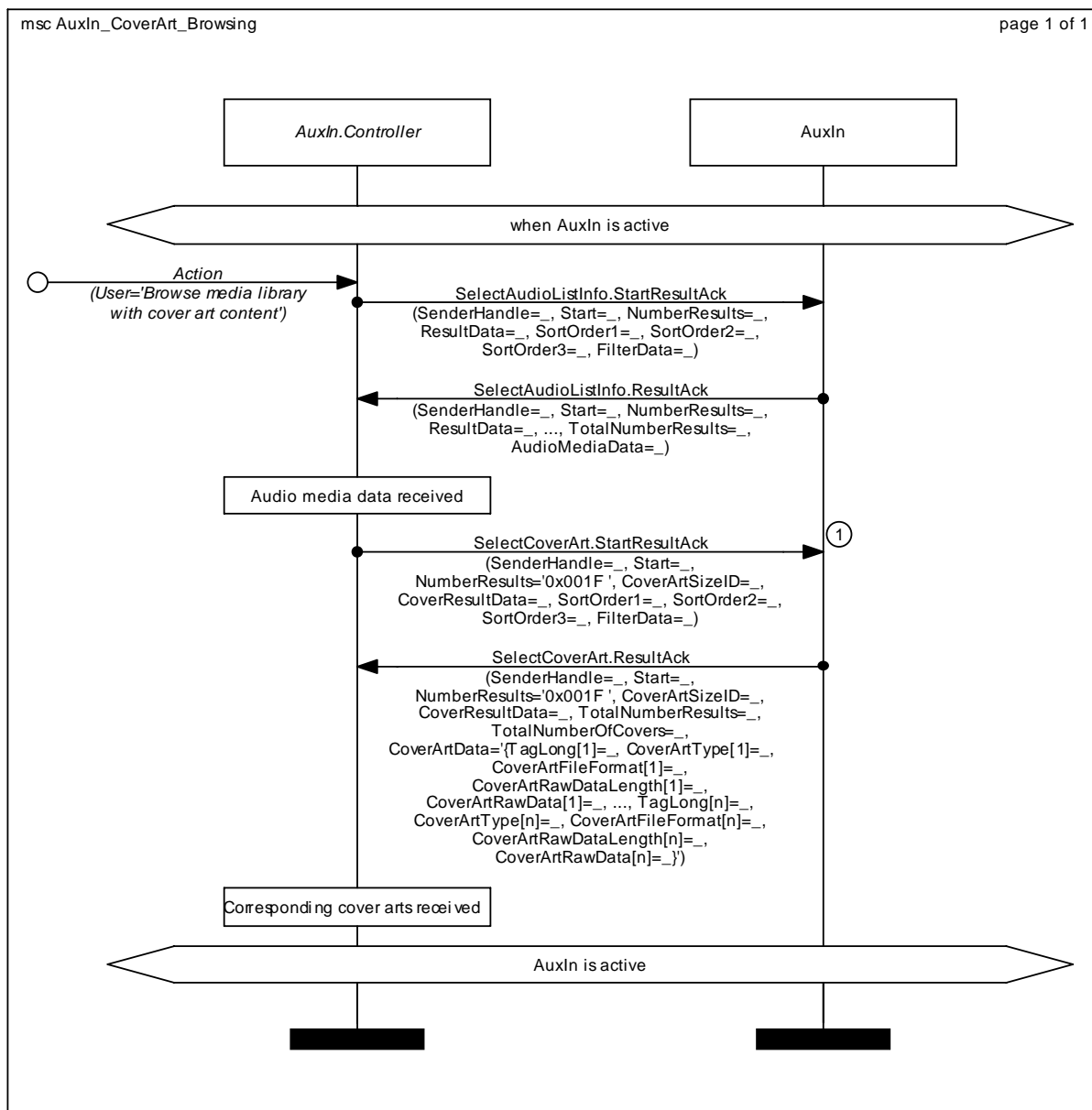
3.5.11 AuxIn_Notify_Time_Position



3.5.12 AuxIn_Notify_Track_Change

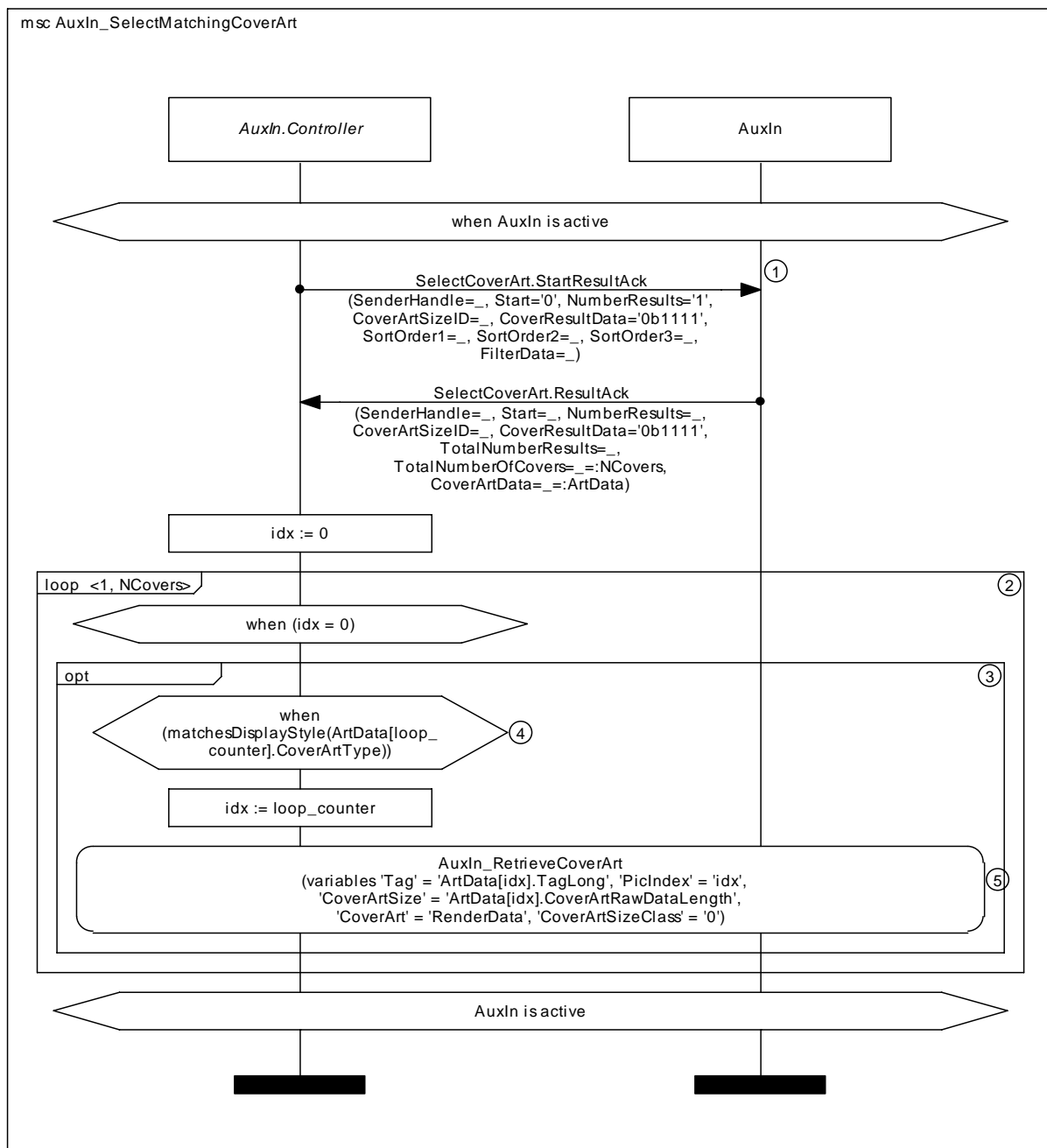


3.5.13 AuxIn_CoverArt_Browsing



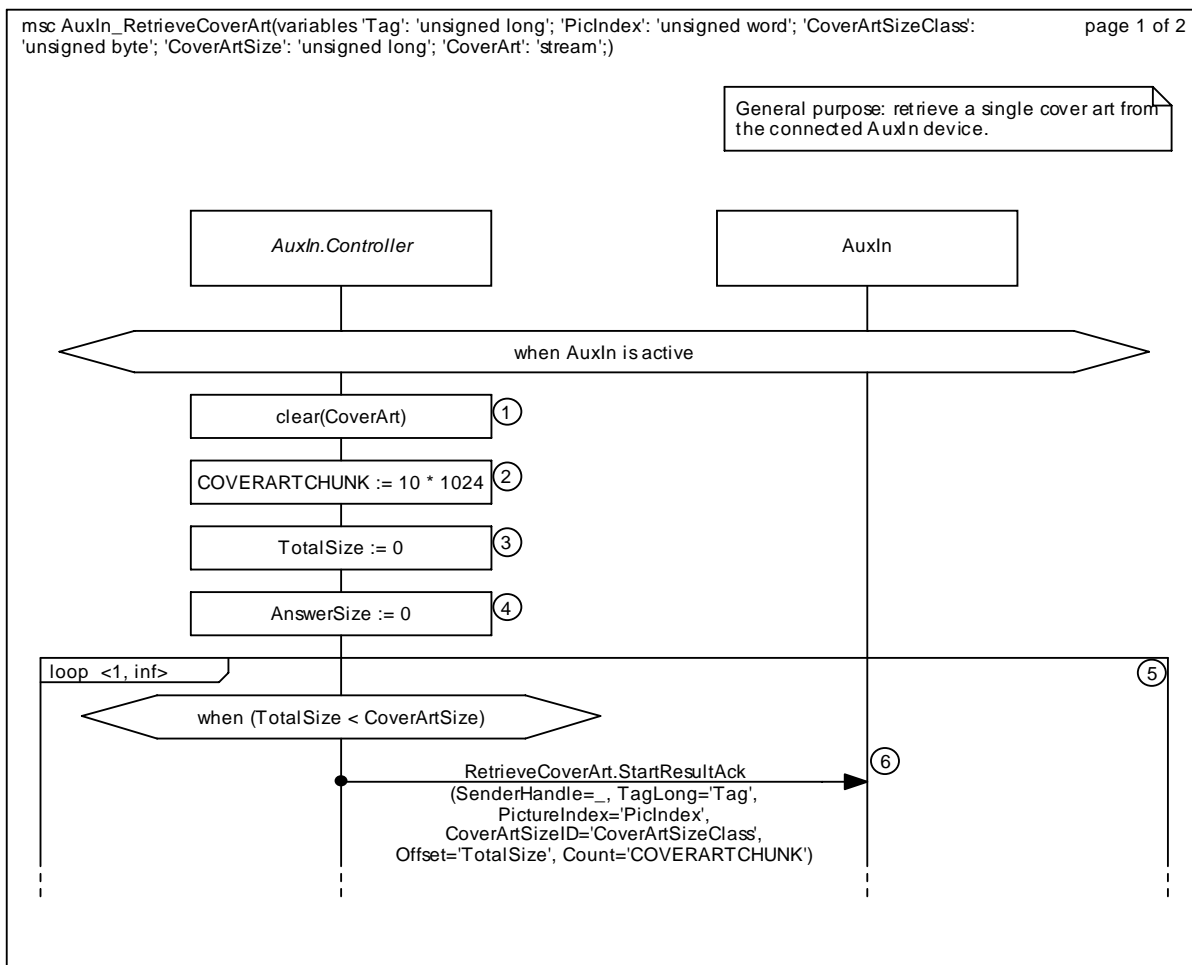
- 1 This function uses the same filter settings of the parameter FilterData as in function SelectAudioListInfo.

3.5.14 AuxIn_SelectMatchingCoverArt

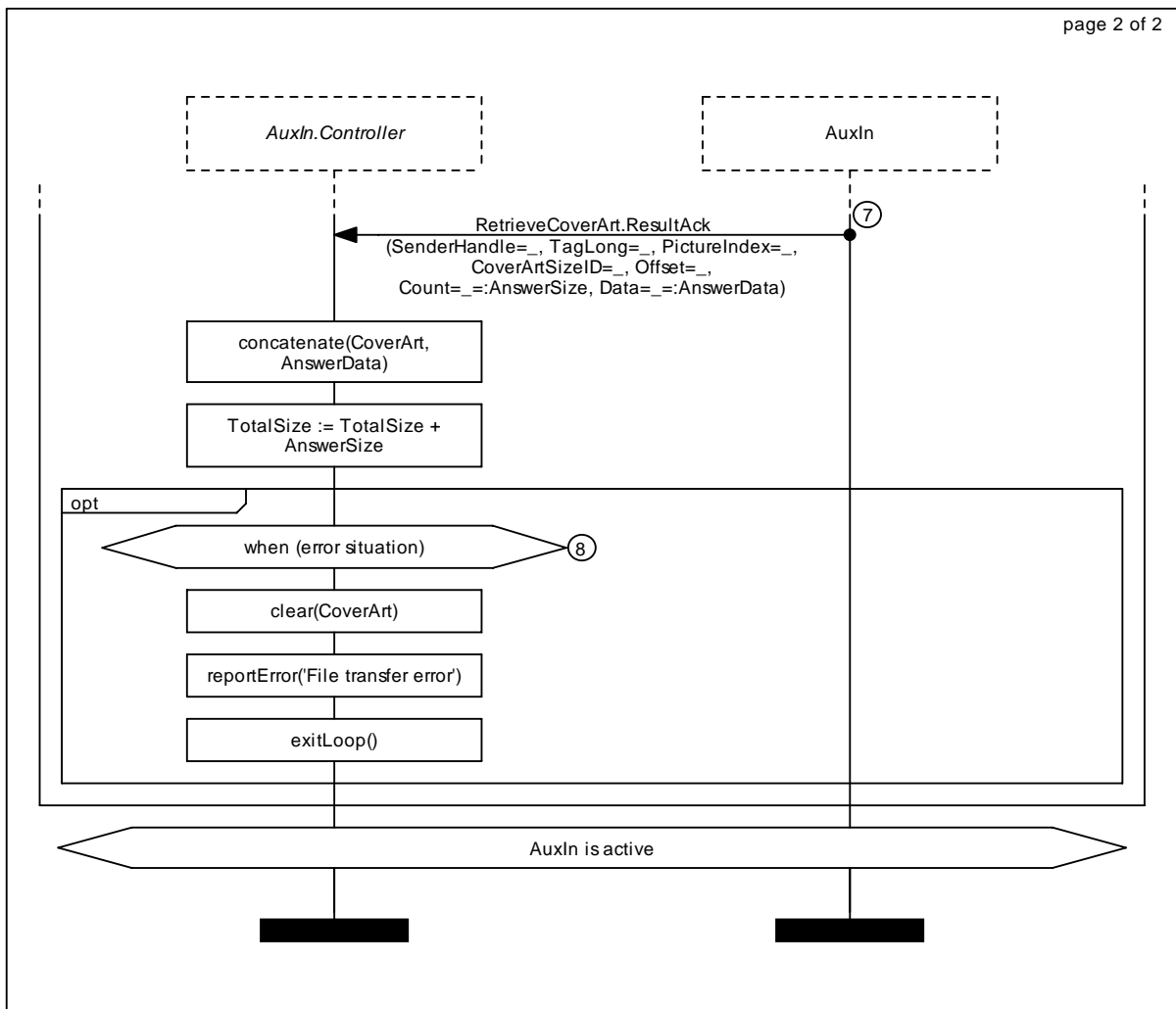


- 1 ask for details of at least TagLong, CoverArtType, CoverArtFileFormat, CoverArtRawDataLength
- 2 search for cover type used for displaying
- 3 retrieve 1st cover art of matching style
- 4 display style may vary, so adjust condition to fit your needs (e.g. CoverArtType = 0x3 for front covers)
- 5 load cover art in its original size; otherwise, change value of CoverArtSizeClass

3.5.15 AuxIn_RetrieveCoverArt



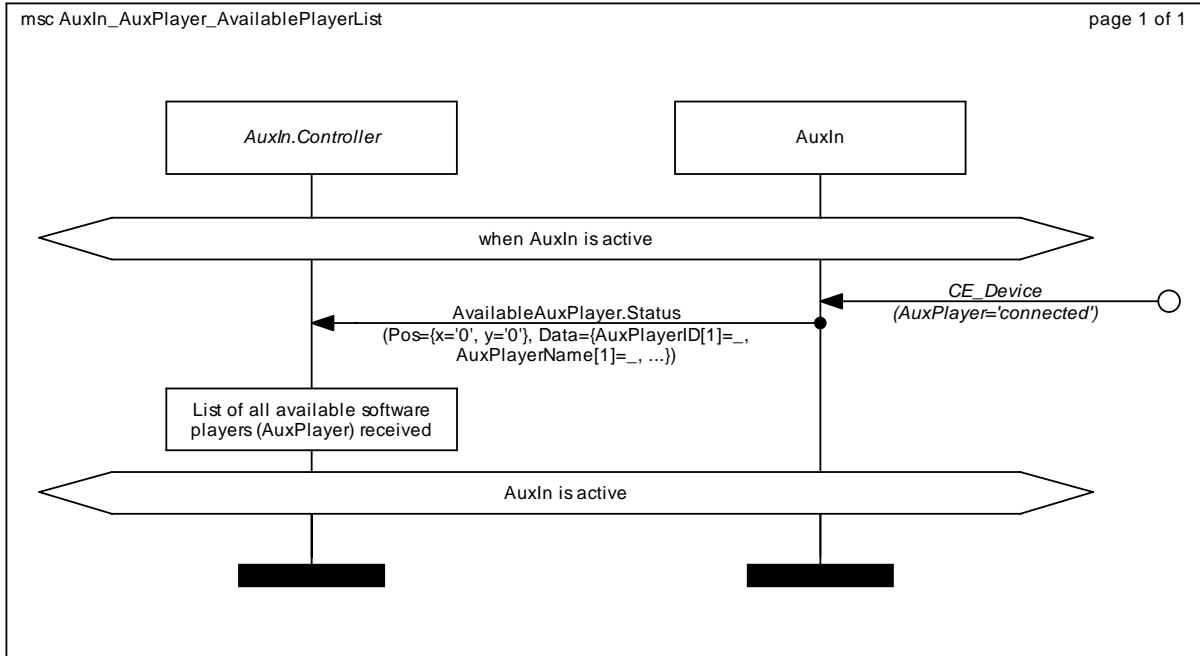
- 1 start with empty cover art;
if no cover art available, report empty data
- 2 constant fitting your transfer block size best
- 3 total amount of bytes received for cover art
- 4 amount of bytes received on last request
- 5 get chunks of cover art;
concatenate chunks in a buffer to return art at end of transmissions
- 6 no special handling for last chunk needed;
the cautious programmer is allowed to change chunk sizes during each iteration of the loop



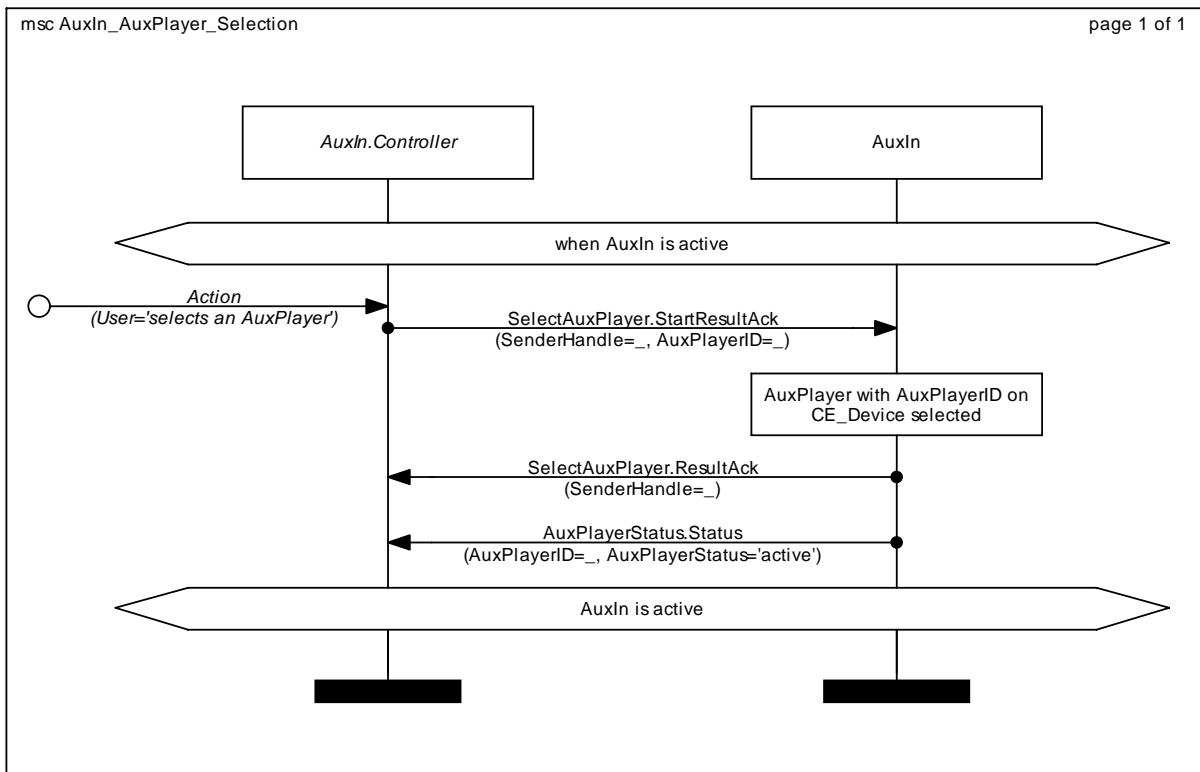
- 7 may contain less data, esp. for last request, which terminates the loop
- 8 e.g. incomplete chunk or timeout on response, even after retry

3.6 AuxPlayer

3.6.1 AuxIn_AuxPlayer_AvailablePlayerList



3.6.2 AuxIn_AuxPlayer_Selection



Notes: