Research & Vehicle Technology
“Infotainment Systems Product Development”

Feature – List Browser Protocol

APIM Infotainment Subsystem Part Specific Specification (SPSS)

Version 1.8.1

UNCONTROLLED COPY IF PRINTED

Version Date: January 23, 2020

FORD CONFIDENTIAL

Revision History

Date	Version	Notes	
May 30, 2013	1.0	Initial Release	
October 14, 2013	1.1		
		LBP-GFUN-294540-2-List Browser Icon and Structured Data	rpaquet2 - Added List structures from A64 document and renamed function. Moved Mobile Apps into Media. No functional changes other then adding new Source and Metadata Icon info for Line In.
		LBP-GREQ-294524-2-Get Item Request	rpaquet2 - Removed SYNC text and replaced with general name List Browser Server.
		LBP-GREQ-294525-2-Audio Source List	rpaquet2 - Change A22j to H22j.
		LBP-GREQ-294526-2-Source Name	rpaquet2 - Removed SYNC text and replaced with general name List Browser Server.
		LBP-GREQ-294527-2-Source Number	rpaquet2 - Removed SYNC text and replaced with general name List Browser Server.
		LBP-GREQ-294528-2-Non Indexing Source	rpaquet2 - Removed SYNC text and replaced with general name List Browser Server.
		LBP-GREQ-294529-2-Unavailable Source	rpaquet2 - Removed SYNC text and replaced with general name List Browser Server.
		LBP-GREQ-294530-2-Non Metadata Source Icon	rpaquet2 - Removed APIM SPSS text from the requirement.
		LBP-GREQ-294531-2-Object State	rpaquet2 - Removed APIM text and replaced with general name List Browser Server. Also removed APIM SPSS text.
		LBP-GREQ-294532-2-Response Time	rpaquet2 - Removed SYNC text and replaced with general name List Browser Server.
		LBP-GREQ-294533-2-Applink Source Unavailable	rpaquet2 -Removed SYNC text and replaced with general name List Browser Server.
		LBP-GREQ-294534-2-No Applink App Selected	rpaquet2 - Removed SYNC text and replaced with general name List Browser Server.
		LBP-GREQ-294535-2-Applink Metadata	rpaquet2 - Removed SYNC text and replaced with general name List Browser Server.
		LBP-GREQ-294539-2-Non Indexing Source with Metadata	rpaquet2 - Removed APIM SPSS text.
		LBP-GREQ-301568-1-Metadata Icons	rpaquet2 - Incorporating A64 document
		LBP-GREQ-301569-1-Source Icons	rpaquet2 - Incorporating A64 document
		LBP-GREQ-301570-1-Phone Icons	rpaquet2 - Incorporating A64 document
		LBP-GREQ-301571-1-Navigation Icons	rpaquet2 - Incorporating A64 document
		LBP-GREQ-301572-1-List Server Generic Media 1 - Media List Structure	rpaquet2 - Incorporating A64 document
		LBP-GREQ-301575-1-List Server Phone Info - Phone List Structure	rpaquet2 - Incorporating A64 document
		LBP-GREQ-301576-1-List Server Navigation Info - Navigation List Structure	rpaquet2 - Incorporating A64 document
		LBP-GREQ-301589-1-List Server Generic List 1 - Sync Services List Structure	rpaquet2 - Incorporating A64 document
		LBP-GREQ-301590-1-List Server Generic List 1 - Valet Mode List Structure	rpaquet2 - Incorporating A64 document
June 13, 2014	1.2		
		LBP-FUN-REQ-019734/B-List Browser Icon and Structured Data (TcSE ROIN-294540-2)	rpaquet2 - Added new Requirements 089676, 089678 and 089684 for GAL/CarPlay.
		LBP-REQ-089676/A-LBP Media definition for GAL/CarPlay.	aaldalla-new requirement for Carplay/GAL
		LBP-REQ-089678/A-LBP phone definition for GAL/CarPlay.	aaldalla-new requirement for GAL/CarPlay
		LBP-REQ-089684/A-LBP Navigation definition for GAL/CarPlay	aaldalla- new requirement for LBP while GAL/CarPlay

December 18, 2015	**1.3**	
	STR-063612/B-List Data Structure (TcSE ROIN-293797-1)+	sberg15: Added new variant of Item Descriptor Attribute requirement
	STR-063612/C-List Data Structure (TcSE ROIN-293797-1)	sberg15: added Activation Event Attribute requirement variant 2.
	LBPv2-SR-REQ-128955/B-Item Descriptor Attribute+	rpaquet2 - Added Radio Mixed Presets to the table. Replaces requirement 19789
	LBPv2-SR-REQ-128955/C-Item Descriptor Attribute	sberg15: Added item descriptor 0x61 Phone Call Category and 0xA4 Radio Source
	STR-063613/B-List Hierarchy (TcSE ROIN-293798-1)	sberg15: added new variant of Available List Servers requirement
	LBPv2-SR-REQ-128954/B-Avaliable List Servers	sberg15: Added information that list server 0x07 RadioDataService1 is used for "Basic" EPG; added new list server 0x09 RadioDataService3 for "Advanced" EPG data. This replaced requirement 19783.
	STR-063617/B-Interface Requirements (TcSE ROIN-293803-1)	sberg15: Added new method descriptions for prefetch and list update function.
	LBPv2-MD-REQ-130601/A-ItemInfo_Rq	New requirement format
	LBPv2-MD-REQ-130388/A-PrefetchState_St	New requirement
	LBPv2-MD-REQ-130602/A-ItemInfo_Rsp	New requirement format
	LBPv2-MD-REQ-130389/A-ListServerUpdate_Ind	New requirement
	STR-063618/B-Navigating Devices and List Servers (TcSE ROIN-293805-1)	sberg15: Added new requirement for Data Prefetch definition.
	LBP-FUN-REQ-019734/C-List Browser Icon and Structured Data (TcSE ROIN-294540-2)+	rpaquet2 - Update requirement 089678.
	LBP-FUN-REQ-019734/D-List Browser Icon and Structured Data (TcSE ROIN-294540-2)	sberg15: added server structure requirements for Basic and Advanced EPG and Journaline to support enhanced DAB features coming with C519
	LBP-REQ-019753/B-Source Icons (TcSE ROIN-301569-1)	sberg15: Separated AM/FM icon ID into a single AM and FM ID; Added icon ID for DAB;
	LBP-REQ-019757/B-List Server Phone Info - Phone List Structure (TcSE ROIN-301575-1)	rpaquet2 - updated table for empty list to fix AIMS issue.
	LBP-REQ-019758/B-List Server Navigation Info - Navigation List Structure (TcSE ROIN-301576-1)	rpaquet2 - Updated No matches found to show Return instead off Ok to fix AIMS issue.
	LBP-REQ-089678/B-LBP Phone definition for GAL/CarPlay.	rpaquet2 - removed text to indicate no phone connected and added CarPlay LBP Structure.
	LBP-REQ-132690/B-List Server RadioDataService1 - EPG List Structure for Centerstack HMI	sberg15: removed Advanced EPG Root server structure.
	LBP-REQ-132691/B-List Server RadioDataService2 - Journaline List Structure for Centerstack HMI	sberg15: Reduced number of list items in root list from 65534 to 51.
	LBP-REQ-192091/A-List Server RadioDataService3 – Advanced EPG List Structure for Centerstack HMI	sberg15: initial release for Advanced EPG
	LBPv2-SR-REQ-205695/A-Activation Event Attribute	sberg15: new variant
	LBP-SR-REQ-129269/A-Data Prefetch	sberg15: new requirement
March 9, 2016	**1.4**	
	STR-063612/D-List Data Structure (TcSE ROIN-293797-1)	rpaquet2 - removed 019787 replaced by 205695 which was already released previously.
	LBPv2-SR-REQ-128955/D-Item Descriptor Attribute	rpaquet2 - Updated 0xA3 Radio Mixed Presets to include a parameter for PresetState. sberg15 - Updated 0xA2 Radio Journaline Data to include a parameter for ChildStatus.
	LBPv2-IIR-REQ-130597/B-LBPClient_LBPServer	rpaquet2 - removed MD 130388 PrefetchState_St.
	STR-063619/B-Functional Definition (TcSE ROIN-293956-1)	rpaquet2 - Added Server list update.
June 1, 2016	**1.5**	
	LBP-FUN-REQ-019734/E-List Browser Icon and Structured Data (TcSE ROIN-294540-2)	rpaquet2 - Added 222052 per Mobile NAV and APIM team request.

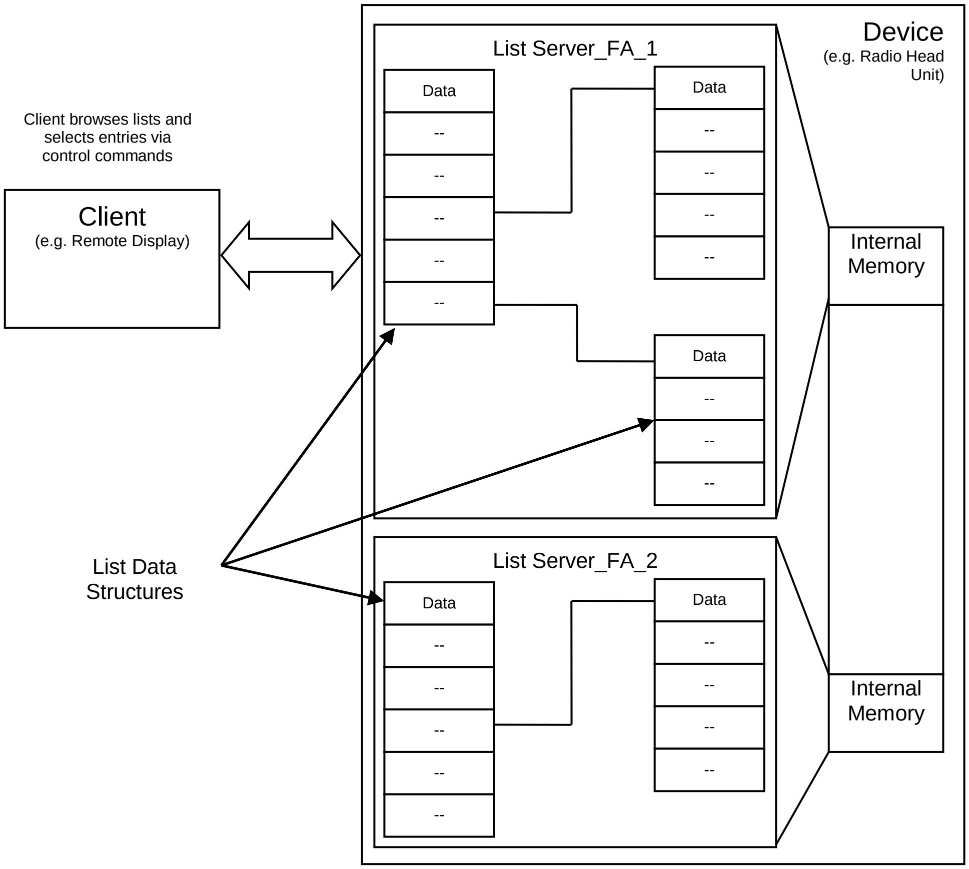
	STR-063529/C-Requirements (TcSE ROIN-294541)	rpaquet2- Added 222052 per Mobile NAV and APIM team request.
	LBP-REQ-222052/A-LBP Navigation definition for Mobile Navigation	rpaquet2 - new requirement for LBP while Mobile NAV active.
February 17, 2017	**1.6**	
	LBPv2-SR-REQ-128954/C-Avaliable List Servers+	rpaquet2 - Added 0x0A Radio 2 for providing source list from AHU to rear display.
	LBPv2-SR-REQ-128954/D-Avaliable List Servers	rpaquet2 - Added new list server for Considerate Prompts POI lists
	LBPv2-MD-REQ-130389/B-ListServerUpdate_Ind	sberg15: corrected typo in ListServer literal All decription field; Added value 0xFFFF All Lists to parameter ListID.
	STR-063618/C-Navigating Devices and List Servers (TcSE ROIN-293805-1)	rpaquet2 - Added Static List Usage requirement needed for Considerate Prompts
	LBP-SR-REQ-250149/A-Static List Usage	rpaquet2: initial release
	LBP-FUR-REQ-019741/B-Unavailable Source (TcSE ROIN-294529-2)	rpaquet2 - Added note to indicate response is for source internal to the responder.
June 16, 2017	**1.7**	
	LBP-REQ-089676/B-LBP Media definition for GAL/CarPlay.	rpaquet2 - Update requirement per Projection mode owners
January 23, 2020	**1.8**	
	LBPv2-SR-REQ-128955/E-Item Descriptor Attribute+	sberg15: added 0xA5 Radio Mixed Station item descriptor
	LBPv2-SR-REQ-128955/F-Item Descriptor Attribute+	sberg15: Updated 0xA5 Radio Mixed Station Item Descriptor with parameters for PI-Code, SCIS, SID and ECC.
	LBPv2-SR-REQ-128955/G-Item Descriptor Attribute	sberg15: Added parameter TPFlag to Item Descriptor 0xA5;
	LBPv2-SR-REQ-128954/E-Avaliable List Servers	sberg15: Added value 0x0C Radio Data Service 4 for Mixed Station List
	LBPv2-MD-REQ-130601/B-ItemInfo_Rq	asimukhi: no content change. Updated Logical-Physical mapping
	LBPv2-MD-REQ-130602/B-ItemInfo_Rsp	asimukhi: no content change. Updated Logical-Physical mapping
	LBPv2-MD-REQ-130389/C-ListServerUpdate_Ind	asimukhi: no content change. Updated Logical-Physical mapping
	LBP-REQ-019756/B-List Server Generic Media 1 - Media List Structure (TcSE ROIN-301572-1)	rpaquet2 - Removed the Non Indexing device sub menu as you can only do a set for Non Indexing devices. Also removed Simialr music from Indexing device list as this was never used.
January 23, 2020	**1.8.1**	
	LBP-REQ-019756/C-List Server Generic Media 1 - Media List Structure (TcSE ROIN-301572-1)	rpaquet2 - Updated to clarify you can only go to list 1 as list 2 for non indexing device was removed.

Table of Contents

1 Architectural Design

The List Browser Protocol is a general way for devices (e.g. radio head unit, remote display) to store and share information across the network. The strategy is based upon devices storing information in a standardized list data structure which can then be used for sharing information between clients and servers. The protocol has been developed to allow any client the ability to browse through and select entries from a list of entries located on any device also implementing this protocol. Lists can be navigated up and down to select entries and traversed forward and backward between parent and child lists. The protocol is flexible in that one common strategy can be applied to browse multiple types of data (e.g. song lists, play lists, audio sources, folders, etc.). The interface can be developed to support the specific data types but can also be flexible and generic to allow new types of data to be accessed without requiring a software update on the client.

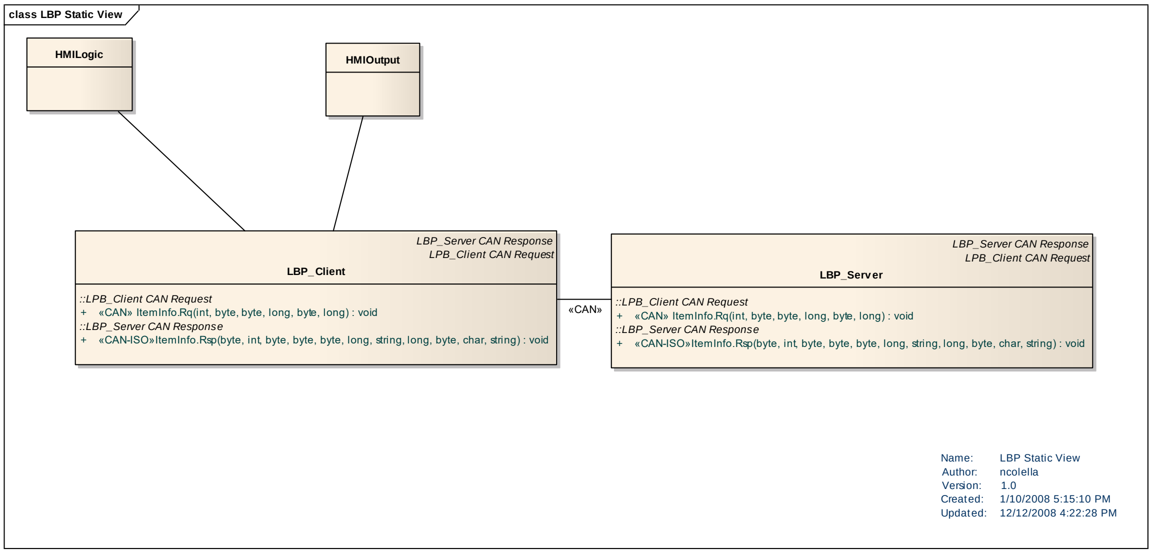
The following figure is a high-level view of the LBP. In the figure a device is shown which contains multiple list severs each dedicated to specific functional area. Other implementation may have one list server dedicated to several functional areas.



This primary section serves to define the underlining strategy of the list browser protocol. The actual usage of the strategy will be defined in other Functional Area Specifications (FAS) with references back to this primary section. The following sub-sections shall provide further descriptions of the list data structure, hierarchy within list and sub-lists, and navigating through lists.

1.1 LBP-SV-REQ-019803/A-Static View (TcSE ROIN-40393-1)

The following is a logical view of the list browser protocol architecture.


class LBP Static View
HMILogic
HMIOutput
LBP_Server CAN Response
LPB_Client CAN Request
LBP_Client
::LPB_Client CAN Request
+ «CAN» ItemInfo.Rq(int, byte, byte, long, byte, long) : void
::LBP_Server CAN Response
+ «CAN-ISO»ItemInfo.Rsp(byte, int, byte, byte, byte, long, string, long, byte, char, string) : void
«CAN»
LBP_Server CAN Response
LPB_Client CAN Request
LBP_Server
::LPB_Client CAN Request
+ «CAN» ItemInfo.Rq(int, byte, byte, long, byte, long) : void
::LBP_Server CAN Response
+ «CAN-ISO»ItemInfo.Rsp(byte, int, byte, byte, byte, long, string, long, byte, char, string) : void
Name: LBP Static View
Author: ncolella
Version: 1.0
Created: 1/10/2008 5:15:10 PM
Updated: 12/12/2008 4:22:28 PM

1.2 LBP-CLD-REQ-019804/A-List Browser Client (TcSE ROIN-159174-1)

The List Brower Client is responsible for addressing list servers and requesting/activating items. The client also is responsible for the HMI output of the information received from the server.

1.3 LBP-CLD-REQ-019805/A-List Browser Server (TcSE ROIN-159175-1)

The List Brower Server is responsible for navigating the lists, keeping track of the active list and the providing information back to the client.

1.4 List Data Structure

The List Data Structure is an information interface which allows a server to present its features and other types of information to clients in a standardized form. By using this protocol and the data structures defined, servers can utilize a standard interface for sharing information with clients.

1.4.1 LBP-SR-REQ-019800/A-List Data Structure (TcSE ROIN-177591-1)

The list data structure is realized as a container for sub-lists and list entries. Each list data structure shall contain the following attributes for each item (list or entry) in a list:

Item Index	Data Type	Activation Event	Object Type	Object State	Item Descriptor
0x0001	0x00-0xFF	Supported/Not Supported	Entry/List	Active/In-Active	{Descriptor Tag}
..	..	..	..	..	..
0xFFFF	..	..	..	..	..

The actual implementation of the storage of the list data structures is left to the device implementer but the method for exchange of information is defined herein.

1.4.2 LBP-SR-REQ-019780/A-Item Index Attribute (TcSE ROIN-40421-1)

The item index is used to denote the position of the item within the list that was specified by the ActiveListID field. Entries shall be numbered so that the first item in a list begins with index number 0x0001, the second position is 0x0002, and so on. The largest addressable item in an ActiveListID is 0xFFFE (65534). 0xFFFF is reserved.

1.4.3 LBP-SR-REQ-019799/A-Object Type Attribute (TcSE ROIN-173633-1)

The Object Type attribute within the list data structure is used to indicate if the object is a "List" object or an "Entry" object. "List" objects are lists that can be navigated via the GetItemInfo.Rq() method or selected for activation via the SetItemInfo.Rq() method. "Entry" objects can only be selected for activation via the SetItemInfo.Rq() method.

1.4.4 LBP-SR-REQ-019801/A-Object State Attribute (TcSE ROIN-177707-1)

The Object State attribute within the list data structure shall be used to indicate if the object is currently activated/in use on the server. This attribute typically pertains to "Entry" objects but may be used by "List" objects.

For example, an entry within a list of song tracks represents a track that is actively playing on the server. When the browsing client requests the list of tracks from the server the server will indicate, via the Object State attribute, that this item is currently active within the list. The client may then provide an indication to the user that this entry is currently active on this server.

1.4.5 LBP-SR-REQ-019781/A-Data Type Attribute (TcSE ROIN-40318-2)

The Data Type attribute within the list data structure is used to denote the information stored in the Item Descriptor attribute for each item. The Data Type attribute shall be used to indicate the type of information the item represents. For example, if the data type is "Album" then the client knows that this item represents the name of an album.

1.4.6 LBPv2-SR-REQ-128955/G-Item Descriptor Attribute

Each item shall also possess the Item Descriptor attribute which can also be used by the client for HMI purposes. The Item Descriptor is typically a large text string which contains a concatenated set of text fields. Each of these fields can be parsed by the client and used for HMI purposes.

The following table defines the link between Data Type Attribute and the Item Descriptor Attribute:

Encoding	Data Type	Item Descriptor	Descriptor Length
0x00	General Reserved		
0x01	General Reserved		
0x02	Generic Text	{Generic Text}	25 characters + EOS
…	General Reserved		
0x1F	General Reserved		
0x20	Media Type	{SourceIcon}{MediaSourceName}	{$0-$FF represents Source icon, $0 = Invalid }{25 characters + EOS}
0x21	Metadata Category	{CatIcon}{CatName}	{$0-$FF represents Category icon, $0 = Invalid}{25 characters + EOS}
…	Media Reserved		
0x3F	Media Reserved		
0x40	Navigation POI	{Direction}{Distance}{POIName}	{$0-$FF Represents direction icon, $0 = Invalid }{8 characters + EOS}{25 characters + EOS} Info: POI name could also be a destination name.

…	Navigation Reserved		
0x5F	Navigation Reserved		
0x60	Phone CallerID	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	{$0-$FF Represents call type icon, $0 = Invalid }{$0-$FF Represents phone type icon, $0 = Invalid }{25 characters + EOS}
0x61	Phone Call Category	{ CallTypeIcon }{CatName}	{$0-$FF Represents call type icon, $0 = Invalid }{25 characters + EOS}
…	Phone Reserved		
0x7F	Phone Reserved		
0x80	BT Device	{DeviceID}{ConnectedIcon}{DeviceName}	{$0-$F Represents the index of the bonded BT device, $0 = Invalid}{$0-$FF Represents the connected icon, $0 = Invalid}{25 characters + EOS}
…	BT Device Reserved		
0x9F	BT Device Reserved		
0xA0	Radio Station	{RadioBand}{IndexNumber}{Frequency}{StationIcon}{StationName}	{$1 = AM, $2 = FM, $3 = DAB, $4 = SDARS, $5 = HD, $0 = Invalid}{$0-$1E Represents the stored station number, $0 = Invalid}{}{$0-$FFFF Represents the frequency, BlockID, SDARS channel number}{$0-$FF Represents a station icon (e.g. HD), $0 = Invalid}{16 characters + EOS}
0xA1	Radio EPG Data	{Icon}{Hour}{Minute}{ProgramName}	{$0-$F Represents EPG program icon, $0 = Invalid }{$0-$17 Represents the hour, $FF = Invalid}{$0-$3B Represents the minute, $FF = Invalid}{128 characters + EOS}
0xA2	Radio Journaline Data	{JournalineListEntry}{ChildStatus}	{50 characters + EOS}{$1 = NotAvailable, $2 = Available, $0 = Invalid}
0xA3	Radio Mixed Presets	{PresetNumber}{PresetState}{RadioBand}{Frequency}{HD Number}{StationNameShort}{StationNameLong}	{$1-$1E, $0 = Invalid}{$0 = Empty, $1 = Available}{$1 = AM, $2 = FM, $3 = DAB, $4 = SDARS, $0 = Invalid} {$0-$FFFF Represents

			the frequency, DAB BlockID, SDARS channel number} {HD Number = $0 - $F} {StationNameShort = 9 characters max + EOS, RDS Markets: PSName = 8 characters max, No PSName = xxx.yyMHz, Non RDS Markets: AM = xxxx kHz, FM = xxx.y MHz, AM HD = AM, FM HD1 = FM, FM HD2+ = xxx.y HDz, SDARS: Shortname = 8 characters max, DAB: ServiceName = 8 characters DAB No Station Name = Blockxxx to APIM and Block xxx to the IPC or MFD} {StationNameLong = DAB: ServiceName = 16 character long name max plus EOS AMFM and SDARS = EOS}
0xA4	Radio Source	{SourceIcon}{RadioSourceName}	{$0-$FF represents Source icon, $0 = Invalid }{25 characters + EOS}
0xA5	Radio Mixed Station	{SourceIcon}{StationNameLong}{StationNameShort}{PI-Code}{SCID}{SID}{ECC}{TPFlag}	{$0-$FF represents Source icon, $0 = Invalid}{StationNameLong = DAB: ServiceName = 16 character long name max plus EOS; AMFM and SDARS = EOS}{StationNameShort = 9 characters max + EOS, RDS Markets: PSName = 8 characters max}{$0-$FFFF represents the PI-Code, $0 = Invalid}{$0-$FFFF represents the SCID, $0 = Invalid}{$0-$FFFF represents the SID, $0 = Invalid}{$0-$FF represents the ECC, $0 = Invalid}{$0 = Invalid, $1 = TRUE, $2 = FALSE}

…	Radio Reserved		
0xAF	Radio Station Reserved		

Note: A list server can contain multiple data types.

1.4.7 LBPv2-SR-REQ-205695/A-Activation Event Attribute

The Activation Event attribute is used to indicate whether an item can be selected or not via the SetItemInfo.Rq() method. Entry Objects that do not support an Activation Event should be indicated as not selectable via the HMI (e.g. empty preset etc.). List Objects may or may not support an Activation Event depending on the server's usage of the object. If a List Object supports an Activation Event it can be selected via SetItemInfo.Rq() method or entered via the GetItemInfo.Rq() method.

1.5 List Hierarchy

This protocol assembles lists, sub-lists, and entry data structures into a hierarchy which is useful when representing information that can be categorized in a hierarchal manner.

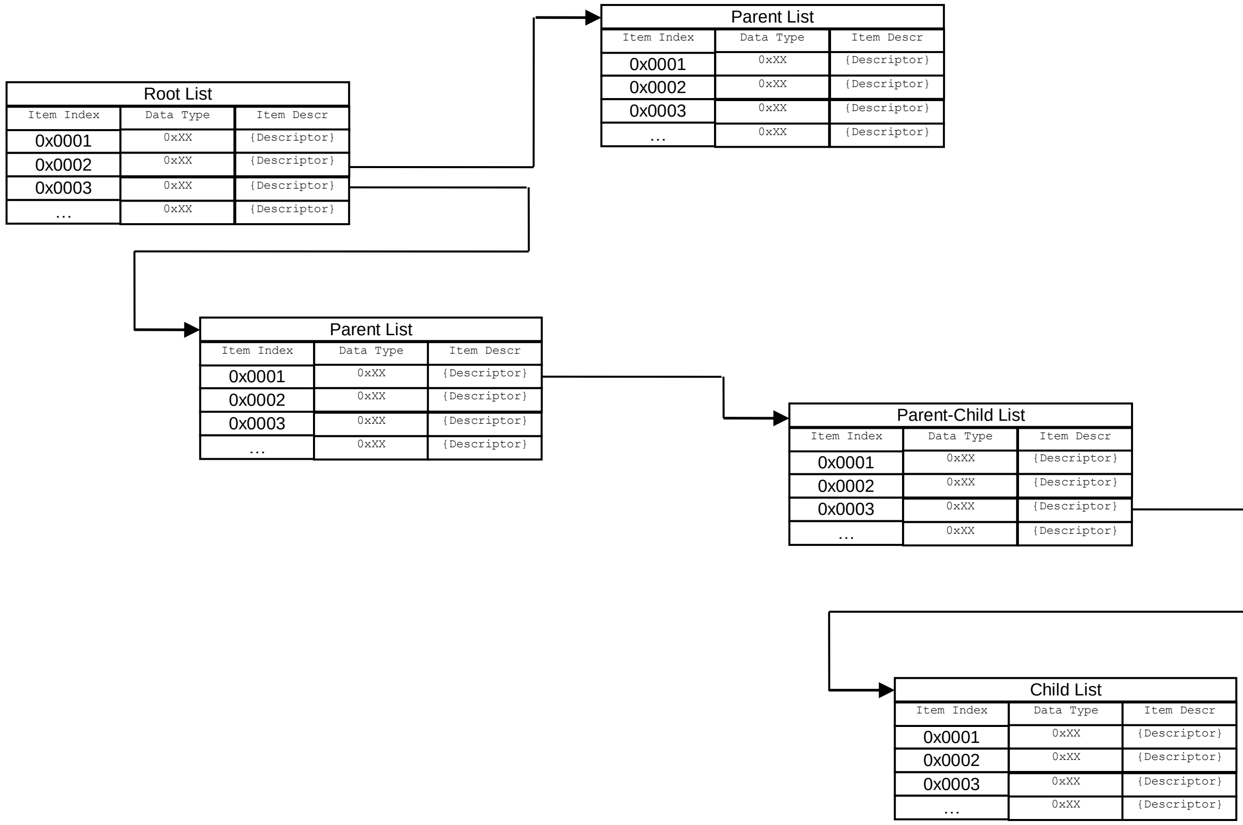
1.5.1 Root List

At the highest level exists the "Root List" and at the minimum the Root List must exist on the server. The contents of the root list are specific to each device. Root Lists can contain sub-lists and root entries.

1.5.2 Parent-Child Lists and List Entries

At the next level exist sub-lists and list entries. List and sub-lists can better be defined as having a parent-child relationship. Therefore, the next levels in the hierarchy can be defined as the Parent lists followed by Child lists. Parent lists can contain child lists and parent entries. Child Lists contain child entries and also sub-lists (i.e. become parents and spawn further child lists).

List Entries exist at the bottom of the hierarchy and are the lowest selectable element which can be traversed in a list (i.e. an entry can not be navigated any further).
The following diagram illustrates the general relationship between the kinds of lists and how they form a hierarchy.



Root List

Item Index	Data Type	Item Descr
0x0001	0xXX	{Descriptor}
0x0002	0xXX	{Descriptor}
0x0003	0xXX	{Descriptor}
…	0xXX	{Descriptor}

Parent List

Item Index	Data Type	Item Descr
0x0001	0xXX	{Descriptor}
0x0002	0xXX	{Descriptor}
0x0003	0xXX	{Descriptor}
…	0xXX	{Descriptor}

Parent List

Item Index	Data Type	Item Descr
0x0001	0xXX	{Descriptor}
0x0002	0xXX	{Descriptor}
0x0003	0xXX	{Descriptor}
…	0xXX	{Descriptor}

Parent-Child List

Item Index	Data Type	Item Descr
0x0001	0xXX	{Descriptor}
0x0002	0xXX	{Descriptor}
0x0003	0xXX	{Descriptor}
…	0xXX	{Descriptor}

Child List

Item Index	Data Type	Item Descr
0x0001	0xXX	{Descriptor}
0x0002	0xXX	{Descriptor}
0x0003	0xXX	{Descriptor}
…	0xXX	{Descriptor}

Figure 1 – Lists-Entry Hierarchy

1.5.3 LBP-SR-REQ-019782/A-List Identifier (TcSE ROIN-40422-2)

A unique list ID, which is assigned by the list server, identifies each list within the list server. The list ID shall be constant during the life of the list.

1.5.4 LBP-SR-REQ-019788/A-Root Index (TcSE ROIN-40430-1)

Each list server shall have a Root List associated to List_ID 0x0000. The Root List shall always be accessible from any point in a list.

1.5.5 LBPv2-SR-REQ-128954/E-Avaliable List Servers

The following are the available addressable list servers:

List Server ID	Name	Comment
0x01	Navigation Info	Used for browsing Nav POIs, previous dest, etc.
0x02	Generic Media 1	Used for browsing media data (e.g. Artist names, titles, etc.)
0x03	Phone Info	Used for browsing phone book, call lists, etc.
0x04	Generic List 1	Contains only items with Data Types 0,1,2.
0x05	Radio 1	Used for browsing radio data (e.g. Preset lists for AM/FM/DAB/SDARS, etc.) started from cluster.
0x06	Remote CD1	Used for browsing media data (e.g. Artist names, titles, etc.)
0x07	Radio Data Service1	Used to transmit DAB data service Basic EPG.

0x08	Radio Data Service2	Used to transmit DAB data service Journaline.
0x09	Radio Data Service3	Used to transmit DAB data service Advanced EPG.
0x0A	Radio 2	Used for browsing radio Source information (e.g. AM,FM,SAT,DAB,CD.) started from Rear Display.
0x0B	Considerate Prompt 1	Static List used to transmit POI list related to Considerate Prompts
0x0C	Radio Data Service4	Used to transmit a mixed station list to the client.
0x0D – 0xFF	Reserved	

1.6 Interface Requirements

1.6.1 LBPv2-IIR-REQ-130597/B-LBPClient_LBPServer

1.6.1.1 LBPv2-MD-REQ-130601/B-ItemInfo_Rq

Message Type: Request

This request signal is used to get or set list content from the list browse server.

Name	Literals	Value	Description
OpCode	-	-	Parameter OpCode is used to distinguish between requesting list data or select an executable item.
	Inactive	0x0	
	GetItemInfo_Rq	0x1	Value GetItemInfo_Rq is used to request list data from a list object.
	SetItemInfo_Rq	0x2	Value SetItemInfo_rq is used by the Client to activate a selected item index during browsing of the device. The value shall only be used on items which support an ActivationEvent. Parameter NbrOfItems and StartItemInd shall be set to 0x0 in this case.
	Reserved	0x3	

SetListServ	-	-	Parameter setListServ is used to address the requested list server.
	Inactive	0x00	
	ServerID_1	0x01	
	…	0x02	
	All	0xFF	Value All is used address all available list servers.

ActiveListID	-	-	Parameter ActiveListID contains the list ID from which the items are being selected. This information is

			provided by the server's response method.
	Root	0x0000	
	List ID1	0x0001	
	….	0x0002	
	List ID 65534	0xFFFE	
	Reserved	0xFFFF	

ItemIndex	-	-	Parameter ItemIndex contains the selected item Index for the requested list.
	BrowseActiveListID	0x0000	Value BrowseActiveListID is used if the client is only browsing up/down the entries of the active list. In this case this value shall be set to 0x0000 (BrowseActiveListID) which indicates to the server to provide the items requested for the active list.
	EntryIndex_1	0x0001	
	….		
	ParentOfActiveListID	0xFFFF	Value ParentOfActiveListID is used if the client needs to retrieve the parent list of the active list. In this case this value shall be set to 0xFFFF (ParentOfActiveListID). This is an indication to the server to navigate to the parent list of the active list and provide that data as requested.

NbrOfItems			Parameter NbrOfItems is used to indicate how many items to provide in the response message.
	Inactive	0x0	
	1 item	0x1	
	…		
	63 items	0x3F	

StartItemInd			Parameter StartItemInd is used to adreess the index value of where to start the request in the response message
	Inactive	0x0000	
	1	0x0001	
	…		
	65534	0xFFFE	
	Reserved	0xFFFF	

1.6.2 LBPv2-IIR-REQ-130599/A-LBPServer_LBPClient

1.6.2.1 LBPv2-MD-REQ-130602/B-ItemInfo_Rsp

Message Type: Response

This response signal is used to transmit the requested list content to the list browse client.

Name	Literals	Value	Description
RspListServ	-	-	Parameter RspListServ is used to indicate which list server is providing the response.
	Inactive	0x00	
	ServerID_1	0x01	
	…		
	Reserved	0xFF	

OpCodeRsp	-	-	Parameter OpCodeRsp is used to return the value of the OpCode associated with the data in the response message.
	Inactive	0x00	
	GetItemInfoRsp	0x01	
	SetItemInfoRsp	0x02	

ActiveListID	-	-	Parameter ActiveListID shall contain the current list ID from which the items are being selected.
	Root	0x0000	
	List ID1	0x0001	
	….	0x0002	
	List ID 65534	0xFFFE	

ParentListID	-	-	The parameter ListID shall contain the parent list ID of the current ActiveListID.
	Root	0x0000	
	List ID1	0x0001	
	….	0x0002	
	List ID 65534	0xFFFE	
	Reserved	0xFFFF	

NbrOfItemsRtn	-	-	Parameter NbrOfItemsRtn is used to indicate how many items provided in the response message
	0	0x0000	
	1	0x0001	
	….		
	30	0x1E	

NbrItemsInSelection			Parameter NbrItemsInSelection is used to indicate the total number of all items in the active list id.
	0	0x0000	
	1	0x0001	
	….		
	65535	0xFFFF	

stringItemContent (Array(1.. NbrOfItemsRtn) of record (ItemIndex, DataType, ActivationEvent, ObjectType, ObjectState, ItemDescriptor)			

ItemIndex	List Title	0x0000	Parameter ItenIndex is used to associate an index to each list item.
	1	0x0001	
	…		
	65535	0xFFFF	

DataType	Reserved	0x00	
	DataTypeID_1	0x01	
	…		
	Reserved	0xFF	

ActivationEvent			Parameter ActivationEvent is used to indicate if a list entry is able to be activated or not.
	Not Supported	0x0	
	Supported	0x1	

ObjectType			Parameter ObjectType is used to indicate if the returned list entry is a List or Entry Object. List objects can contain additional data, entry objects can be executed.
	List Label	0x0	
	Entry Object	0x1	
	List Object	0x2	

ObjectState			The parameter ObjectState is used to indicate if the returned list entry is the active on the server or not.
	Inactive	0x0	
	Active	0x1	

ItemDescriptor			The parameter Item Descriptor is typically a large text string which contains a concatenated set of text

			fields. Each of these fields can be parsed by the client and used for HMI purposes.
	{Descriptor Tag}		

Example ItemContent:

Item Index	Data Type	ActivationEvent	ObjectType	Item Descriptor
0x0000	0x02	No	List Label	{Generic Text}
0x0001	0x00-0xFE	Yes/No	..	{Descriptor Tag}
..	..	..	..	
0xFFFE	..	..	..	

1.6.2.1.1 TP-LOG-TPL-REQ-023169/C-SID-76-LBP1_ItemInfo_Rsp (TcSE ROIN-159709-6)

Data size: up to Variable (Coding Table I/Coding Table II) bytes

Byte 0: Signal identifier

0x76: LBP1_ItemInfo_Rsp

Byte 1: Utilization

0x01 Radio_Service1	– Radio General (AM, FM, AST, DAB, SDARS)
0x02 Radio_Service2	– SDARS
0x03 Radio_Service3	– DAB
0x11 MP_Media1	– CD
0x12 MP_Media2	– BT Audio Streaming
0x13 MP_Media3	– USB
0x14 MP_Media4	– iPod
0x17 MP_Media7	– Generic Metadata
0x22 Nav_Service2	– Navigation
0x31 MobileCom_Service1	– Mobile Phone
0x74: DataService4	– List Browser Data

Byte 2: Command Execution Status

0x0y: Final Result	–	Success
0x1y: Final Result	–	Fail
0x2y: Final Result	–	Information
0x3y: Intermediate Result	–	Wait

Byte 3: Character Coding

Bit 0-5: Reserved

Bit 6-7: Coding

0x0: Coding Table I
0x0000-0xFFFF UNICODE UTF-16 (2 byte per char)
0x1: Coding Table II
0x00-0xFF Latin-9 (1 byte per char)

Byte 4: OpCodeRsp:

Bit 0-5: reserved

Bit 6 - 7: OpCodeRsp

0x0: Inactive
0x1: GetItemInfoRsp
0x2: SetItemInfoRsp
0x3: Reserved

Byte 5: *RspListServ :*

0x00: Inactive
0x01: ServerID_1
...

0xFF: Reserved

Byte 6-7: ActiveListID

0x0000: Root
0x0001: ListID_1
0x0002: ListID_2
....
0xFFFE:
0xFFFF: Reserved

Byte 8-9: ParentListID

0x0000: Root
0x0001: ListID_1
0x0002: ListID_2
....
0xFFFE:
0xFFFF: Reserved

Byte 10: NbrOfItemsRtn

0x00: Reserved
0x01: 1
0x02: 2
....
0xFE: 254
0xFF: Reserved

Byte 11-12: NbrOfItemsInSelection

0x0000: Reserved
0x0001:
0x0002:
....
0xFFFE:
0xFFFF:

Byte 13 up to Variable (Coding Table I/Coding Table II): Channel Info

Array(1..NumberOfItemsRtn) of record (ItemIndex, DataType, ObjectType, ObjectState, ActivationEvent, ItemDescriptor)

Record definition (up to Variable (Coding Table I/Coding Table II) bytes):
Byte 0-1: ItemIndex
0x0000: List Label
0x0001
..
0xFFFF

Byte 2: DataType
0x00
0x01
..
0xFF

Byte 3:
Bit 0 - 1: Reserved
Bit 2 - 3: *ObjectType*
0x0: List Label
0x1: Entry ~~List~~
0x2: List~~Entry~~
Bit 4 - 5: *ObjectState*
0x0: Inactive
0x1: Active

Bit 6 - 7: *ActivationEvent*
0x0: Not Supported
0x1: Supported

Byte 4 up to Byte Variable: ItemDescriptor
{Descriptor Tag} – Refer to descriptor table and DataType.

1.6.2.2 LBPv2-MD-REQ-130389/C-ListServerUpdate_Ind

Message Type: Indication

This indication method is used to indicate a server list update to the client. This indication can be used to start fetching the updated list content. This method shall not be used during initial server list build up.

Name	Literals	Value	Description
ListServer	-	-	Parameter ListServer is used to address the list server which is/was updated.
	Inactive	0x00	
	ServerID_1	0x01	
	…	0x02	
	All	0xFF	Value All is used to address all available list servers.

ListID	-	-	The parameter ListID is used to indicate which server list is getting updated.
	Root	0x0	Root List is getting updated
	ListID1	0x0001	ListID1 is getting updated
	ListID2	0x0002	ListID2 is getting updated
	…		
	List ID65534	0xFFFE	ListID65534 is getting updated
	All Lists	0xFFFF	Value All Lists is used to indicate that all Lists within a server have been changed.

ParentID	-	-	The parameter ParentID is used to indicate the parent list which contains the updating ListID.
	Root	0x0	Root List is ParentID
	ListID1	0x0001	ListID1 is ParentID
	ListID2	0x0002	ListID2 is ParentID
	…		
	List ID65534	0xFFFE	ListID65534 is ParentID
	Reserved	0xFFFF	

EntryIndex	-	-	The parameter EntryIndex shall be used to indicate which list entry from the ParentID is getting updated. This could be used to indicate this update via HMI output (e.g. grey out list entry etc.)

	NoDataExists	0x0000	NoDataExisis shall be set if the root list is getting updated.
	EntryIndex1	0x0001	Entry 1 from parent list is getting updated.
	EntryIndex2	0x0002	Entry 2 from parent list is getting updated.
	…		
	EntryIndex65534	0xFFFE	Entry 65534 from parent list is getting updated.
	Reserved	0xFFFF	
ListStatus	-	-	The parameter is used to transmit the different states of the server list.
	Inactive	0x0	
	Valid	0x1	Valid is set if the list is updated and ready.
	Updating	0x2	Updating is set if the list is getting updated with new data.
	Reserved	0x3	

1.7 Navigating Devices and List Servers

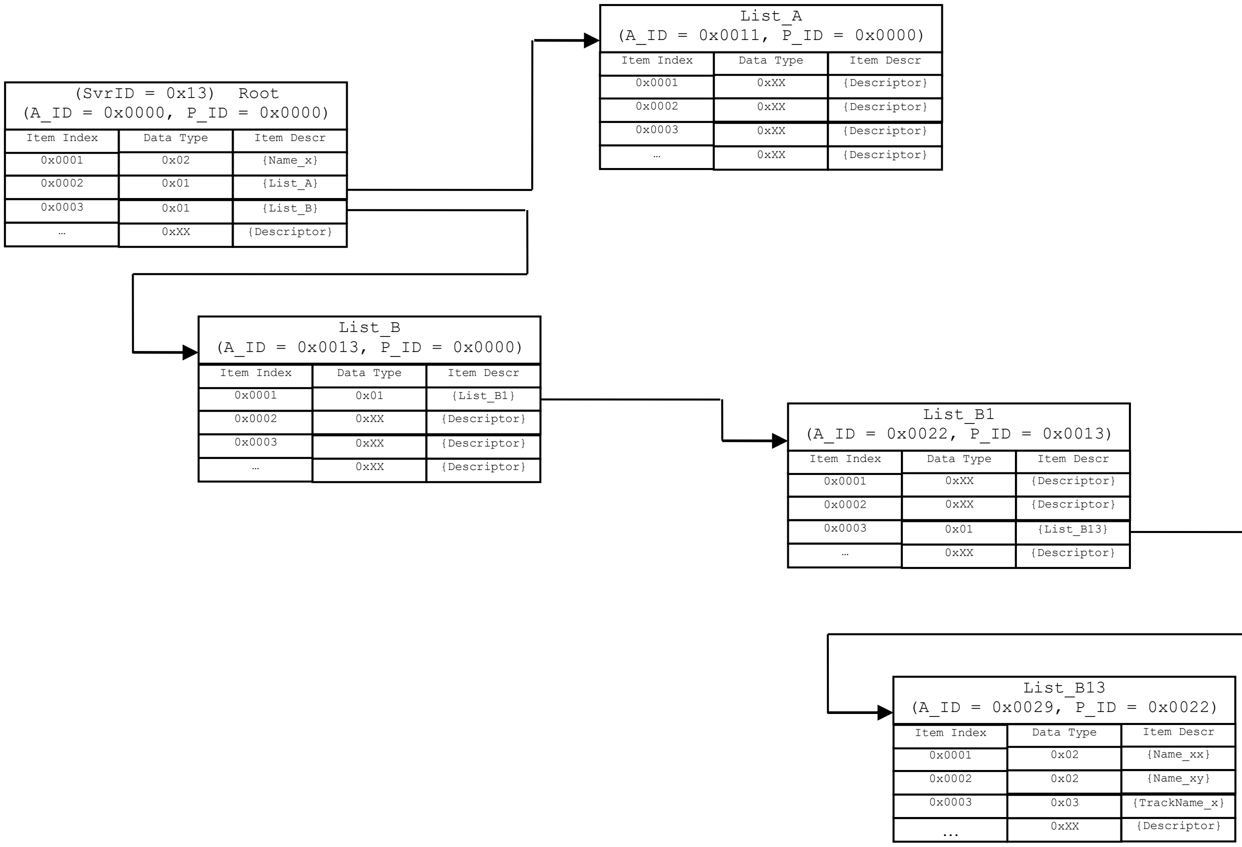
The process of an HMI client browsing the information available on a particular device begins with the client first accessing a particular list server present on the device. Once an HMI client has accessed a particular list server, the client can scroll up and down through items in the list and then request an entry in the list. The client can also move forward and backward between lists and sub-lists by requesting the particular (parent or child) list from the list server.

Several methods are provided for navigating the hierarchy and retrieval of information, GetItemInfo.Request(), SetItemInfo.Request(), and GeItemtInfo.Response(). The client utilizes the GetItemInfo.Request() methods for requesting item information and utilizes the SetItemInfo.Request for item activation. The server utilizes the GetItemInfo.Response() method for responding to information requests. The details of these methods are defined in the Interface Requirements portion of this section.

1.7.1 LBP-SR-REQ-019786/A-Requesting List Contents (TcSE ROIN-40428-1)

The browsing client shall utilize the GetItemInfo.Rq() method for managing the retrieval of data from a particular list server. To browse through a device via the HMI, the client must address a particular list server. For example, to view the contents of a USB device, the client shall target the USB list server and based on the list data structures (retrieved earlier) stored on this server the HMI can then browse through parent-child lists (e.g. artists, albums, etc.).

To support an example, imagine a server contains the following items:



(SvrID = 0x13) Root
(A_ID = 0x0000, P_ID = 0x0000)

Item Index	Data Type	Item Descr
0x0001	0x02	{Name_x}
0x0002	0x01	{List_A}
0x0003	0x01	{List_B}
…	0xXX	{Descriptor}

List_A
(A_ID = 0x0011, P_ID = 0x0000)

Item Index	Data Type	Item Descr
0x0001	0xXX	{Descriptor}
0x0002	0xXX	{Descriptor}
0x0003	0xXX	{Descriptor}
…	0xXX	{Descriptor}

List_B
(A_ID = 0x0013, P_ID = 0x0000)

Item Index	Data Type	Item Descr
0x0001	0x01	{List_B1}
0x0002	0xXX	{Descriptor}
0x0003	0xXX	{Descriptor}
…	0xXX	{Descriptor}

List_B1
(A_ID = 0x0022, P_ID = 0x0013)

Item Index	Data Type	Item Descr
0x0001	0xXX	{Descriptor}
0x0002	0xXX	{Descriptor}
0x0003	0x01	{List_B13}
…	0xXX	{Descriptor}

List_B13
(A_ID = 0x0029, P_ID = 0x0022)

Item Index	Data Type	Item Descr
0x0001	0x02	{Name_xx}
0x0002	0x02	{Name_xy}
0x0003	0x03	{TrackName_x}
…	0xXX	{Descriptor}

Where A_ID is the ActiveListID and P_ID is the ParentListID of the ActiveList. If the client is in the root list and wishes to request the contents of item 3 (List_B) within the root list, then the request parameters would be set to the following, ActiveListID = 0x0000 (root is being browsed), ItemIndex = 0x0003 (item selected), NumberOfItems = 0x05 (number of entries requested), StartItemInd = 0x0001 (starting index of entries to return).

The response back from the server would be A_ID = 0x0013, P_ID = 0x0000 with the data structure depicted for List_B in the figure above.

If the client now wishes to select item 1 (List_B1) within the active sub-list, then the request parameters would be set to the following, ActiveListID = 0x0013 (active list), ItemIndex = 0x0001 (item selected), NumberOfItems = 0x05 (number of entries requested), StartItemInd = 0x0001 (starting index of entries to return)..

The response back from the server would be A_ID = 0x0022, P_ID = 0x0013 with the data structure depicted for List_B1 in the figure above.

1.7.2 LBP-SR-REQ-019790/A-Selecting List Entry (TcSE ROIN-40732-2)

The browsing client shall utilize the SetItemInfo.Rq() method for activation of a particular item within the ActiveListID. This method can only be used by items which support ActivationEvents.

To support the example described previously: If the active list is reported A_ID = 0x0022, P_ID = 0x0013 with the data structure depicted for List_B1 in the figure above. If the client now wishes to select and activate item 3, assuming the ActivationEvent is supported for this item, within the active sub-list, then the settings for the SetItemInfo.Rq() method would be as follows: ActiveListID = 0x0022 (active list), ItemIndex = 0x0003 (item selected).

1.7.3 LBP-SR-REQ-019791/A-Tracking (TcSE ROIN-41670-1)

The server shall be responsible for tracking the navigation between the lists.

1.7.4 LBP-SR-REQ-019784/A-SetItem - Audio Resource Request (TcSE ROIN-40314-1)

If a SetItemInfo.Rq() requires a change to the audio resources, the server shall be responsible for issuing the request for connection of the respective audio resource.

1.7.5 LBP-SR-REQ-019785/A-Label of List (TcSE ROIN-40317-2)

Each list within a list server shall have an associated textual label/title. The server shall provide the label of each list to the client.

Within GetItemInfo.Rsp(), the ItemIndex equal to 0x0000 shall be used to contain the label of the active list. The Data Type for the label shall be set equal to 0x02 (Generic text). The ActivationEvent shall be set to "No" for labels. The Object Type shall be set to 0x00 (List Label).

The label shall be provided when the StartItemIndex of GetItemInfo.Rq() is equal to 0x0001.
The label shall not be counted as a member of NbrItemsInSelection.

The label shall be counted as a member of NbrOfItemsRtn by both the client and the server. Therefore, when the client requests information starting at index 0x0001and would like five items of information returned, the client must request NbrOfItems equal to six. Likewise the server will respond back with NbrOfItemsRtn equal to six. With a starting index greater then 0x0001 with five items requested, the client would request NbrOfItems equal to five with the server responding back with NbrOfItemsRtn equal to five.

1.7.6 LBP-SR-REQ-019795/A-Client requests invalid parent-child list (TcSE ROIN-31400-1)

If the client requests a parent-child that is either out of range or does not exist, the server shall respond back with CES = 0x14 (Final Result – Failure, requested index out of range).

1.7.7 LBP-SR-REQ-019796/A-Client selects invalid entry (TcSE ROIN-31407-1)

If the client selects invalid entry that is either out of range or does not exist, the server shall respond back with CES = 0x14 (Final Result – Failure, requested index out of range).

1.7.8 LBP-SR-REQ-019797/A-Client selects invalid parent-child list (TcSE ROIN-31414-1)

If the client selects a parent-child that is either out of range or does not exist, the server shall respond back with CES = 0x14 (Final Result – Failure, requested index out of range).

1.7.9 LBP-SR-REQ-019798/A-SetItem - Server Response (TcSE ROIN-160332-1)

Upon reception of a valid SetItemInfo.Rq(), the server shall provide a response with CES = 0x01 (Final Result – Success). All fields after the CES field will not be transmitted.

1.7.10 LBP-SR-REQ-129269/A-Data Prefetch

While the List Browse Client is in prefetch state then the List Browse Server shall ignore the activation event attribute if the client requests for list objects which have an activation event supported. This is required to avoid e.g. activation of a radio band when requesting its list content.

1.7.11 LBP-SR-REQ-250149/A-Static List Usage

A static list is defined as a list that cannot change the order of the original structure. Once created the Client can directly request for one of the items in the list knowing that the items will always be the in the same location..

The static list can have new items added into new ItemIndex locations and once added cannot be removed.

Example: A static root list contains 3 possible selections. The Client sends a GetItemInfo_Rq for ItemIndex 3 of the root list without having to ask for the root list first. The Server will respond with the list for ItemIndex 3.

2 Functional Definition

2.1 LBP-FUN-REQ-019707/A-Request Root List (TcSE ROIN-293807-1)

2.1.1 Use Cases

2.1.1.1 LBP-UC-REQ-019708/A-Request Root List (TcSE ROIN-292216-1)

Actors	System
Pre-conditions	Infotainment Network is active.
Scenario Description	Client requests Root List from server.
Post-conditions	Root List is retrieved from server.
List of Exception Use Cases	N/A
Interfaces	G-HMI, Vehicle System Interface

2.1.2 White Box View

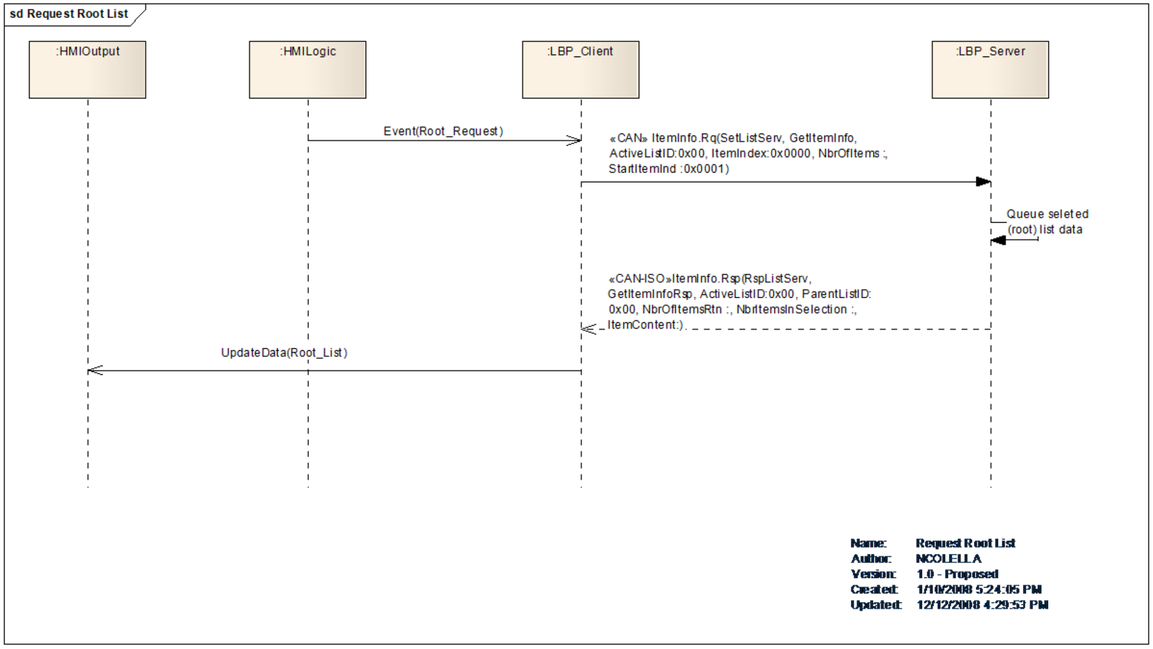
2.1.2.1 White Box Scenarios

2.1.2.1.1 LBP-SD-REQ-019709/A-Request Root List from List Server (TcSE ROIN-39796-1)

Linked Elements

LBP-UC-REQ-019762/A-Request Root List (TcSE ROIN-30390-1)

Sequence Diagram


sd Request Root List
:HMIOutput
:HMILogic
:LBP_Client
:LBP_Server
Event(Root_Request)
«CAN» ItemInfo.Rq(SetListServ, GetItemInfo, ActiveListID:0x00, ItemIndex:0x0000, NbrOfItems :, StartItemInd :0x0001)
Queue seleted (root) list data
«CAN-ISO»ItemInfo.Rsp(RspListServ, GetItemInfoRsp, ActiveListID:0x00, ParentListID: 0x00, NbrOfItemsRtn :, NbrItemsInSelection :, ItemContent:)
UpdateData(Root_List)
Name: Request Root List
Author: NCOLELLA
Version: 1.0 - Proposed
Created: 1/10/2008 5:24:05 PM
Updated: 12/12/2008 4:29:53 PM

2.2 LBP-FUN-REQ-019710/A-Browse a parent List from Root List (TcSE ROIN-293810-1)

2.2.1 Use Cases

2.2.1.1 LBP-UC-REQ-019711/A-Browse a Parent List from the Root List (TcSE ROIN-292217-1)

Actors	System
Pre-conditions	Infotainment Network is active. The root list has been obtained from the server. Root list contains parent lists.
Scenario Description	The client requests a parent list within the root list for browsing. For example, the root list contains several folders and the user selects a folder to browse/read its contents.
Post-conditions	Server responds with requested parent list structure.
List of Exception Use Cases	N/A
Interfaces	G-HMI, Vehicle System Interface

2.2.2 White Box View

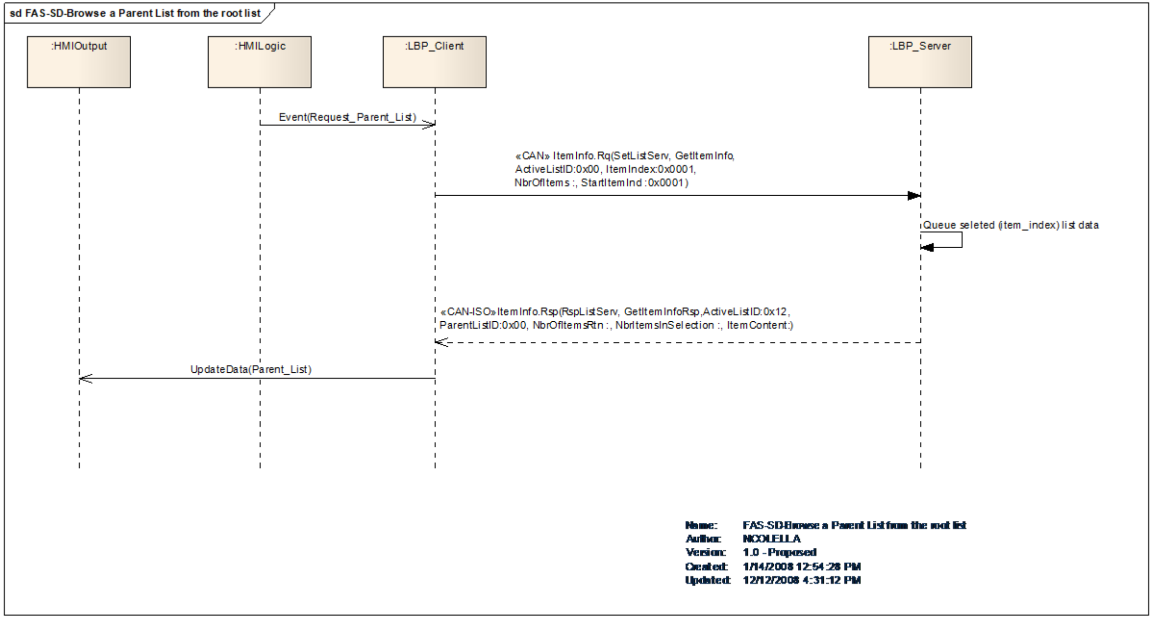
2.2.2.1 White Box Scenarios

2.2.2.1.1 LBP-SD-REQ-019712/A-Browse a Parent List from the root list (TcSE ROIN-39798-1)

Linked Elements

LBP-UC-REQ-019764/A-Browse a Parent List from the Root List (TcSE ROIN-30397-1)

Sequence Diagram


sd FAS-SD-Browse a Parent List from the root list
:HMIOutput
:HMILogic
:LBP_Client
:LBP_Server
Event(Request_Parent_List)
«CAN» ItemInfo.Rq(SetListServ, GetItemInfo, ActiveListID:0x00, ItemIndex:0x0001, NbrOfItems :, StartItemInd :0x0001)
Queue seleted (item_index) list data
«CAN-ISO»ItemInfo.Rsp(RspListServ, GetItemInfoRsp,ActiveListID:0x12, ParentListID:0x00, NbrOfItemsRtn :, NbrItemsInSelection :, ItemContent:)
UpdateData(Parent_List)
Name: FAS-SD-Browse a Parent List from the root list
Author: NCOLELLA
Version: 1.0 - Proposed
Created: 1/14/2008 12:54:28 PM
Updated: 12/12/2008 4:31:12 PM

2.3 LBP-FUN-REQ-019713/A-Selecting an Entry from Root List (TcSE ROIN-293813-1)

2.3.1 Use Cases

2.3.1.1 LBP-UC-REQ-019714/A-Selecting an Entry from the Root List (TcSE ROIN-292218-1)

Actors	System
Pre-conditions	Infotainment Network is active.

	The root list has been obtained from the server. The root list contains selectable entries.
Scenario Description	The client selects an entry from the root list.
Post-conditions	Server responds with activation of the selected entry.
List of Exception Use Cases	N/A
Interfaces	G-HMI, Vehicle System Interface

2.3.2 White Box View

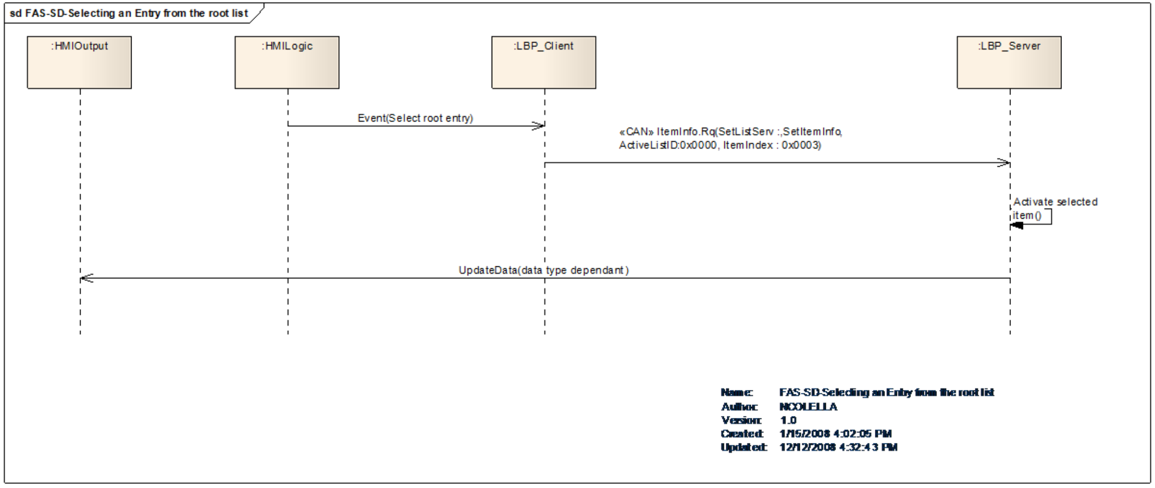
2.3.2.1 White Box Scenarios

2.3.2.1.1 LBP-SD-REQ-019715/A-Selecting an Entry from the root list (TcSE ROIN-39799-1)

Linked Elements

LBP-UC-REQ-019766/A-Selecting a Entry from the Root List (TcSE ROIN-30421-1)

Sequence Diagram


sd FAS-SD-Selecting an Entry from the root list
:HMIOutput
:HMILogic
:LBP_Client
:LBP_Server
Event(Select root entry)
«CAN» ItemInfo.Rq(SetListServ :,SetItemInfo, ActiveListID:0x0000, ItemIndex : 0x0003)
Activate selected item()
UpdateData(data type dependant)
Name: FAS-SD-Selecting an Entry from the root list
Author: NICOLELLA
Version: 1.0
Created: 1/15/2008 4:02:05 PM
Updated: 12/12/2008 4:32:43 PM

2.4 LBP-FUN-REQ-019716/A-Browsing Child List from Parent List (TcSE ROIN-293816-1)

2.4.1 Use Cases

2.4.1.1 LBP-UC-REQ-019717/A-Browsing a Child List from a Parent List (TcSE ROIN-292219-1)

Actors	System
Pre-conditions	Infotainment Network is active. The client is browsing a parent list which contains child lists.
Scenario Description	The client requests a child list from within a parent list. For example, the user is browsing Folder_A (parent) which contains several folders (children). The user selects one of the child folders for browsing.
Post-conditions	The server responds with contents of selected child list.
List of Exception Use Cases	N/A
Interfaces	G-HMI, Vehicle System Interface

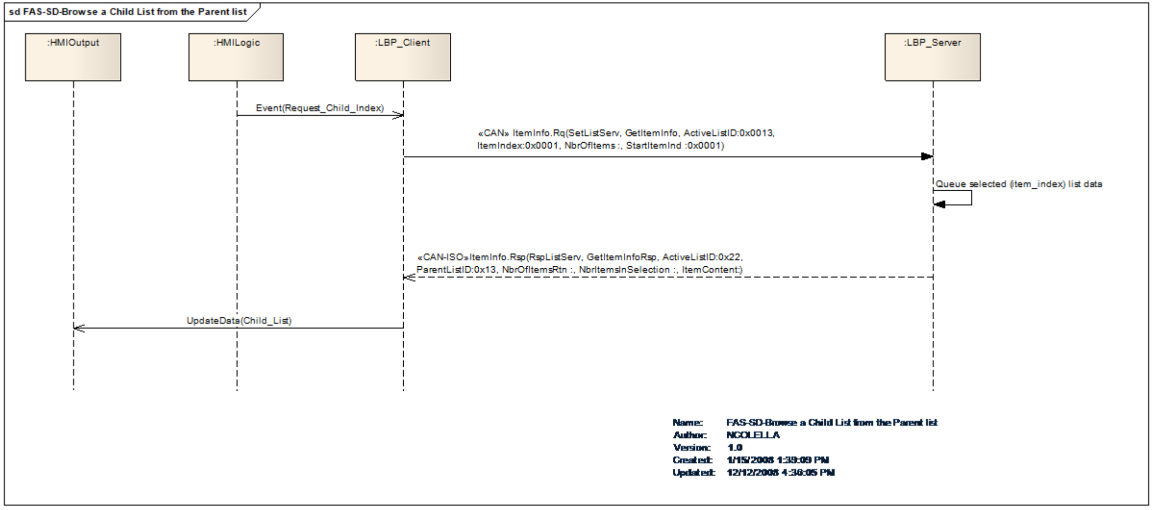
2.4.2 White Box View

2.4.2.1 White Box Scenarios

2.4.2.1.1 LBP-SD-REQ-019718/A-Browse a Child List from the Parent list (TcSE ROIN-39800-1)

Linked Elements
LBP-UC-REQ-019768/A-Browsing a Child List from a Parent List (TcSE ROIN-30427-1)

Sequence Diagram


sd FAS-SD-Browse a Child List from the Parent list
:HMIOutput
:HMILogic
:LBP_Client
:LBP_Server
Event(Request_Child_Index)
«CAN» ItemInfo.Rq(SetListServ, GetItemInfo, ActiveListID:0x0013, ItemIndex:0x0001, NbrOfItems :, StartItemInd :0x0001)
Queue selected (item_index) list data
«CAN-ISO»ItemInfo.Rsp(RspListServ, GetItemInfoRsp, ActiveListID:0x22, ParentListID:0x13, NbrOfItemsRtn :, NbrItemsInSelection :, ItemContent:)
UpdateData(Child_List)
Name: FAS-SD-Browse a Child List from the Parent list
Author: NCOLELLA
Version: 1.0
Created: 1/15/2008 1:39:09 PM
Updated: 12/12/2008 4:36:05 PM

2.5 LBP-FUN-REQ-019719/A-Selecting and Entry from a parent/Child List (TcSE ROIN-293819-1)

2.5.1 Use Cases

2.5.1.1 LBP-UC-REQ-019720/A-Selecting an Entry from a Parent-Child List (TcSE ROIN-292220-1)

Actors	System
Pre-conditions	Infotainment Network is active. The parent-child list contains selectable entries.
Scenario Description	The client selects an entry from the parent-child list.
Post-conditions	Server responds with activation of the selected entry.
List of Exception Use Cases	N/A
Interfaces	G-HMI, Vehicle System Interface

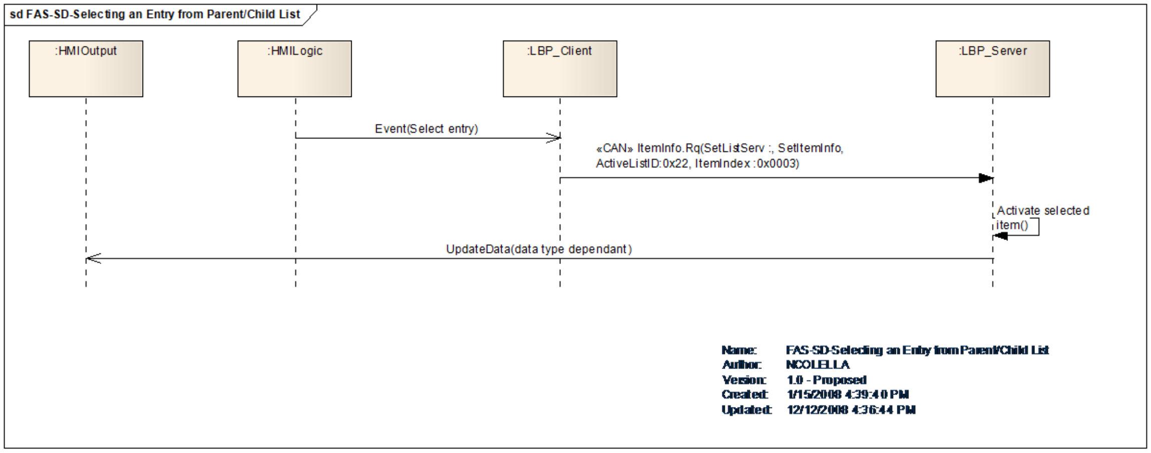
2.5.2 White Box View

2.5.2.1 White Box Scenarios

2.5.2.1.1 LBP-SD-REQ-019721/A-Selecting an Entry from Parent/Child List (TcSE ROIN-39801-1)

Linked Elements
LBP-UC-REQ-019770/A-Selecting an Entry from a Parent-Child List (TcSE ROIN-30439-1)

Sequence Diagram


sd FAS-SD-Selecting an Entry from Parent/Child List
:HMIOutput
:HMILogic
:LBP_Client
:LBP_Server
Event(Select entry)
«CAN» ItemInfo.Rq(SetListServ :, SetItemInfo, ActiveListID:0x22, ItemIndex :0x0003)
Activate selected item()
UpdateData(data type dependant)
Name: FAS-SD-Selecting an Entry from Parent/Child List
Author: NCOLELLA
Version: 1.0 - Proposed
Created: 1/15/2008 4:39:40 PM
Updated: 12/12/2008 4:36:44 PM

2.6 LBP-FUN-REQ-019722/A-Traversing up the Hierarchy from child to parent list (TcSE ROIN-293822-1)

2.6.1 Use Cases

2.6.1.1 LBP-UC-REQ-019723/A-Traversing up the Hierarchy from Child to Parent (TcSE ROIN-292221-1)

Actors	System
Pre-conditions	Infotainment Network is active. Client currently browsing a Child List.
Scenario Description	The client requests the parent list of the current child list. For example, the user is browsing Folder_B which is a child of Folder_A and wishes to view the content list of Folder_A.
Post-conditions	Server responds back with the Parent List of the current Child List.
List of Exception Use Cases	N/A
Interfaces	G-HMI, Vehicle System Interface

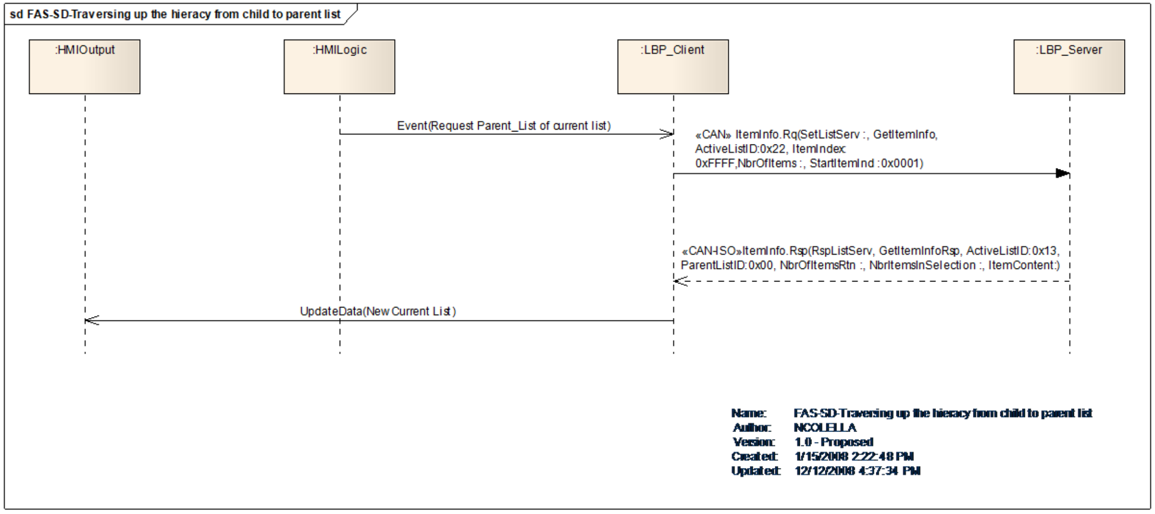
2.6.2 White Box View

2.6.2.1 White Box Scenarios

2.6.2.1.1 LBP-SD-REQ-019724/A-Traversing up the hieracy from child to parent list (TcSE ROIN-39802-1)

Linked Elements
LBP-UC-REQ-019772/A-Traversing up the hierarchy from Child to Parent (TcSE ROIN-30445-1)

Sequence Diagram


sd FAS-SD-Traversing up the hieracy from child to parent list
:HMIOutput
:HMILogic
:LBP_Client
:LBP_Server
Event(Request Parent_List of current list)
«CAN» ItemInfo.Rq(SetListServ :, GetItemInfo, ActiveListID:0x22, ItemIndex: 0xFFFF,NbrOfItems :, StartItemInd :0x0001)
«CAN-ISO»ItemInfo.Rsp(RspListServ, GetItemInfoRsp, ActiveListID:0x13, ParentListID:0x00, NbrOfItemsRtn :, NbrItemsInSelection :, ItemContent:)
UpdateData(New Current List)
Name: FAS-SD-Traversing up the hieracy from child to parent list
Author: NICOLELLA
Version: 1.0 - Proposed
Created: 1/15/2008 2:22:48 PM
Updated: 12/12/2008 4:37:34 PM

2.7 LBP-FUN-REQ-019725/A-Traversing up the Hierarchy from Parent to Root (TcSE ROIN-293825-1)

2.7.1 Use Cases

2.7.1.1 LBP-UC-REQ-019726/A-Traversing up the Hierarchy from Parent to Root (TcSE ROIN-292222-1)

Actors	System
Pre-conditions	Infotainment Network is active. Client currently browsing a Parent List.
Scenario Description	The client requests the root list of the current parent list. For example, the user is browsing Folder_A which is a Parent in the Root List and wishes to view the content list of the Root list.
Post-conditions	Server responds back with the Root List of the current Parent list.
List of Exception Use Cases	N/A
Interfaces	G-HMI, Vehicle System Interface

2.7.2 White Box View

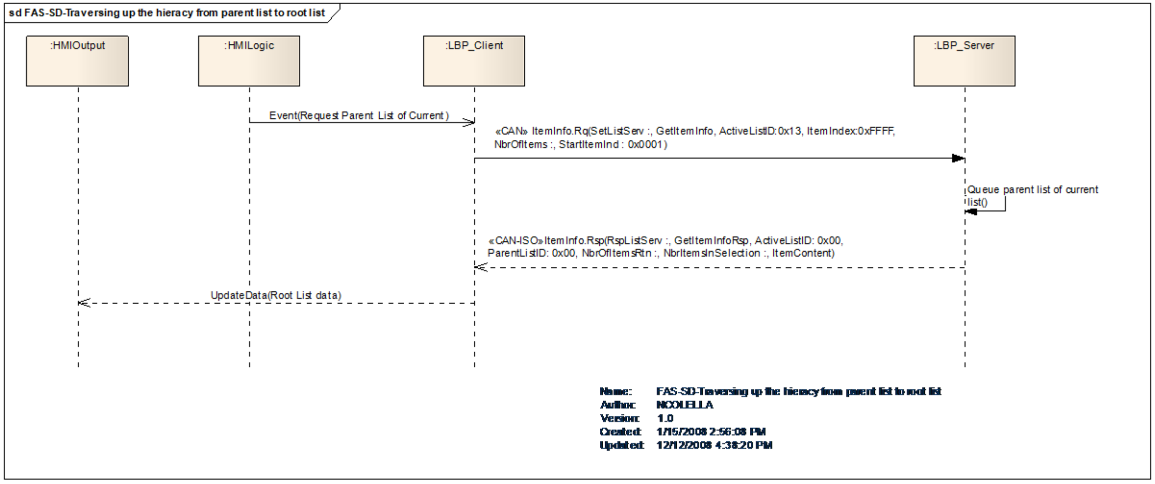
2.7.2.1 White Box Scenarios

2.7.2.1.1 LBP-SD-REQ-019727/A-Traversing up the hieracy from parent list to root list (TcSE ROIN-39810-1)

Linked Elements

LBP-UC-REQ-019774/A-Traversing up the hierarchy from Parent to Root (TcSE ROIN-30451-1)

Sequence Diagram


sd FAS-SD-Traversing up the hieracy from parent list to root list
:HMIOutput
:HMILogic
:LBP_Client
:LBP_Server
Event(Request Parent List of Current)
«CAN» ItemInfo.Rq(SetListServ :, GetItemInfo, ActiveListID:0x13, ItemIndex:0xFFFF, NbrOfItems :, StartItemInd : 0x0001)
Queue parent list of current list()
«CAN-ISO» ItemInfo.Rsp(RspListServ :, GetItemInfoRsp, ActiveListID: 0x00, ParentListID: 0x00, NbrOfItemsRtn :, NbrItemsInSelection :, ItemContent)
UpdateData(Root List data)
Name: FAS-SD-Traversing up the hieracy from parent list to root list
Author: NICOLELLA
Version: 1.0
Created: 1/15/2008 2:56:08 PM
Updated: 12/12/2008 4:38:20 PM

2.8 LBP-FUN-REQ-019728/A-Browsing down a list (TcSE ROIN-293828-1)

2.8.1 Use Cases

2.8.1.1 LBP-UC-REQ-019729/A-Browsing down a List (TcSE ROIN-292223-1)

Actors	System
Pre-conditions	Infotainment Network is active. Client is currently browsing a list.
Scenario Description	The client is browsing a list and requests the next index of entries.
Post-conditions	Server responds back with requested index of entries.
List of Exception Use Cases	N/A
Interfaces	G-HMI, Vehicle System Interface

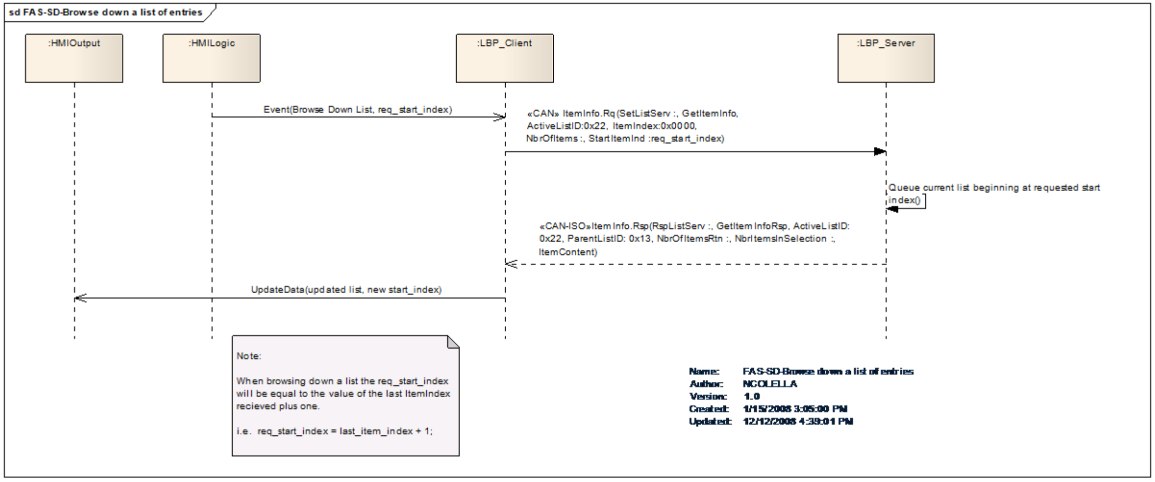
2.8.2 White Box View

2.8.2.1 White Box Scenarios

2.8.2.1.1 LBP-SD-REQ-019730/A-Browse down a list of entries (TcSE ROIN-39811-1)

Linked Elements
LBP-UC-REQ-019776/A-Browsing down a list (TcSE ROIN-30463-1)

Sequence Diagram


sd FAS-SD-Browse down a list of entries
:HMIOutput
:HMILogic
:LBP_Client
:LBP_Server
Event(Browse Down List, req_start_index)
«CAN» ItemInfo.Rq (SetListServ :, GetItemInfo, ActiveListID:0x22, ItemIndex:0x0000, NbrOfItems :, StartItemInd :req_start_index)
Queue current list beginning at requested start index()
«CAN-ISO»ItemInfo.Rsp(RspListServ :, GetItemInfoRsp, ActiveListID: 0x22, ParentListID: 0x13, NbrOfItemsRtn :, NbrItemsInSelection :, ItemContent)
UpdateData(updated list, new start_index)

Note:

When browsing down a list the req_start_index will be equal to the value of the last ItemIndex recieved plus one.

i.e. req_start_index = last_item_index + 1;

Name:	FAS-SD-Browse down a list of entries
Author:	NCOLELLA
Version:	1.0
Created:	1/15/2008 3:05:00 PM
Updated:	12/12/2008 4:39:01 PM

2.9 LBP-FUN-REQ-019731/A-Browsing up a list (TcSE ROIN-293831-1)

2.9.1 Use Cases

2.9.1.1 LBP-UC-REQ-019732/A-Browsing up a List (TcSE ROIN-292224-1)

Actors	System
Pre-conditions	Infotainment Network is active. Client is currently browsing a list.
Scenario Description	The client is browsing a list and requests the previous index of entries.
Post-conditions	Server responds back with requested index of entries.
List of Exception Use Cases	N/A
Interfaces	G-HMI, Vehicle System Interface

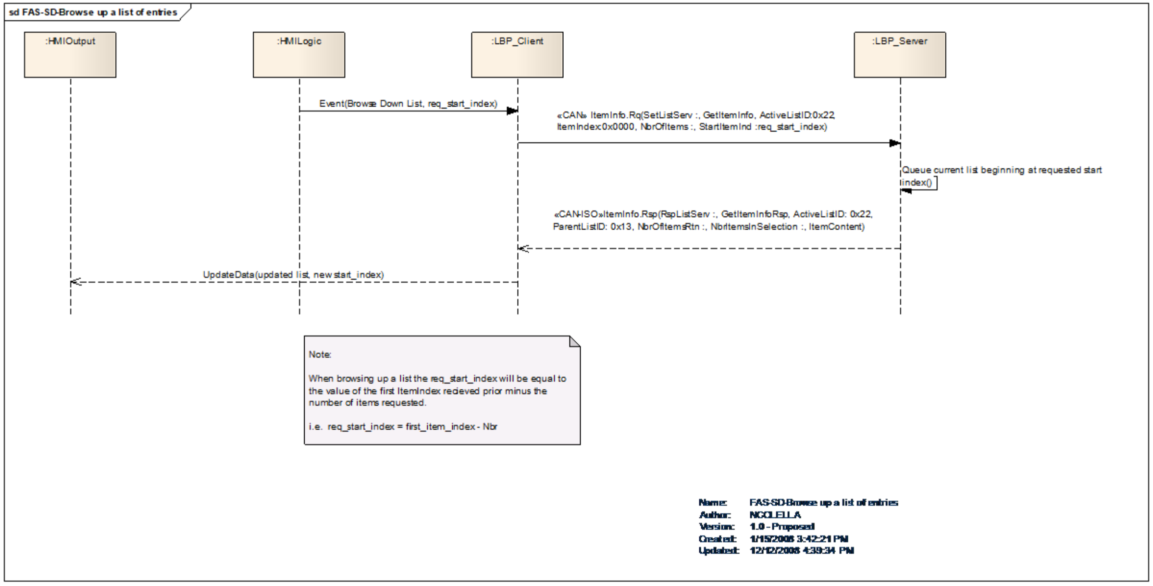
2.9.2 White Box View

2.9.2.1 White Box Scenarios

2.9.2.1.1 LBP-SD-REQ-019733/A-Browse up a list of entries (TcSE ROIN-39812-1)

Linked Elements
LBP-UC-REQ-019778/A-Browsing up a list (TcSE ROIN-30469-1)

Sequence Diagram


sd FAS-SD-Browse up a list of entries
:HMIOutput
:HMILogic
:LBP_Client
:LBP_Server
Event(Browse Down List, req_start_index)
«CAN» ItemInfo.Rq(SetListServ :, GetItemInfo, ActiveListID:0x22, ItemIndex:0x0000, NbrOfItems :, StartItemInd :req_start_index)
Queue current list beginning at requested start index()
«CAN-ISO»ItemInfo.Rsp(RspListServ :, GetItemInfoRsp, ActiveListID: 0x22, ParentListID: 0x13, NbrOfItemsRtn :, NbrItemsInSelection :, ItemContent)
UpdateData(updated list, new start_index)
Note:
When browsing up a list the req_start_index will be equal to the value of the first ItemIndex recieved prior minus the number of items requested.
i.e. req_start_index = first_item_index - Nbr
Name: FAS-SD-Browse up a list of entries
Author: NCOLELLA
Version: 1.0 - Proposed
Created: 1/15/2008 3:42:21 PM
Updated: 12/12/2008 4:39:34 PM

2.10 LBP-FUN-REQ-019734/E-List Browser Icon and Structured Data (TcSE ROIN-294540-2)

2.10.1 Requirements

2.10.1.1 LBP-FUR-REQ-019735/A-Translations (TcSE ROIN-294523-1)

All text strings must reference the MUI database for full language support and match the current system language as selected in the System Settings menu.

2.10.1.2 LBP-FUR-REQ-019736/A-Get Item Request (TcSE ROIN-294524-2)

If the List Browser Server receives a getitem request of 0xFFFF then return to the parent menu. Refer to SPSS Requirment LBP-GREQ-41673-1 for further clarification.

2.10.1.3 LBP-FUR-REQ-019737/A-Audio Source List (TcSE ROIN-294525-2)

The list of audio sources shall be as follows: USB (1, 2...n), BT Stereo (1, 2...n), {app name} (1, 2...n), SD Card (1, 2...n), AV In/Line In. Note: For the source order, reference the H22j specification.

2.10.1.4 LBP-FUR-REQ-019738/A-Source Name (TcSE ROIN-294526-2)

The {source name} shall be the name of the selected List Browser Server audio source.

2.10.1.5 LBP-FUR-REQ-019739/A-Source Number (TcSE ROIN-294527-2)

The {source #} shall be the number of the List Browser Server audio source; for example 2:USB would be the second USB source in the list.

2.10.1.6 LBP-FUR-REQ-019740/A-Non Indexing Source (TcSE ROIN-294528-2)

If the device is non indexing, The List Browser Server must return a one list item of {Play All}.

2.10.1.7 LBP-FUR-REQ-019741/B-Unavailable Source (TcSE ROIN-294529-2)

If the getitem is requesting an unavailable source, the List Browser Server shall return a Command Execution Status of 0x15 (Final Result-Failure, connected environment not reacting). Refer to requirement TP-GREQ-138094-2-CES.

Note: This requirement applies to sources internal to the responding Server ID (sources contained in the request Module). No response required for Server ID which is requested outside of the module.

2.10.1.8 LBP-FUR-REQ-019742/A-Non Metadata Source Icon (TcSE ROIN-294530-2)

If NonMetadataSrc is Yes, then MetadataIcon_1 must still be populated with the corresponding icon from the Metadata Icons table. Only Bluetooth, A/V In/Line In and Non Indexing Source may be used if NonMetadataSrc is Yes. Refer to requirement TP-LOF-GTPC-160692-2-SID-79-MediaInformation_St.

2.10.1.9 LBP-FUR-REQ-019743/A-Object State (TcSE ROIN-294531-2)

When the List Browser Server is returning a list for Media sources, the Object State must be given. For example if the USB source is active and you are browsing the media source list, Object State must be Active. Refer to requirement LBP-GREQ-177707-Object State Attribute.

2.10.1.10 LBP-FUR-REQ-019744/A-Response Time (TcSE ROIN-294532-2)

The List Browser Server must respond to a List Browser Client request in <100ms with the fully populated list i.e. the user should never see the 'Please Wait' while navigating through the cluster menus. The Navigation POI list (Active List0x0005) is the only exception and must have a fully populated list in < 5 seconds.

2.10.1.11 LBP-FUR-REQ-019745/A-Applink Source Unavailable (TcSE ROIN-294533-2)

If the audio source is currently unavailable (i.e. USB device is not connected or Applink media is not detected) The List Browser Server must not send the source information to the cluster. The only source that must always be sent is A/V In/Line In.

2.10.1.12 LBP-FUR-REQ-019746/A-No Applink App Selected (TcSE ROIN-294534-2)

If Applink media is detected and no app has been selected, the List Browser Server must not send the source information to the cluster.

2.10.1.13 LBP-FUR-REQ-019747/A-Applink Metadata (TcSE ROIN-294535-2)

If Applink media is detected and an app is currently selected, the List Borowser Server must send the {mainField 1} and {mainField 2} metadata to the cluster in the MediaInformation_St TP protocol.

2.10.1.14 LBP-FUR-REQ-019748/A-Applink Metadata Icon (TcSE ROIN-294536-1)

The Metadata Icon fields for all Mobile Apps must be set to 0x17 (No Icon).

2.10.1.15 LBP-FUR-REQ-019749/A-Total Play Time (TcSE ROIN-294537-1)

The TotalPlayTime in message 0x3F0 must be set to 0 for all Mobile Apps.

2.10.1.16 LBP-FUR-REQ-019750/A-Applink App Name (TcSE ROIN-294538-1)

If the Applink media is detected and an app is currently selected (i.e. Pandora), the Item Descriptor field must fill in with the currently selected application i.e. {0x07}{Pandora}.

2.10.1.17 LBP-FUR-REQ-019751/A-Non Indexing Source with Metadata (TcSE ROIN-294539-2)

If a Non Indexing source (i.e. Bluetooth) provides metadata, NonMetadataSrc must be set to No and artist/title info with corresponding metadata icons must be sent to the cluster. Refer to requirment TP-LOF-GTPC-160692-2-SID-79-MediaInformation_St.

2.10.1.18 LBP-REQ-019752/A-Metadata Icons (TcSE ROIN-301568-1)

Icon Name	Icon ID	MetadataIcon_1/ MetadataIcon_2	NonMetadataSrc	Comments(Used For)
Invalid	0	0x00	No	
Song Title	1	0x01	No	Music Title, Rhapsody Channel Title
Music Artist	2	0x02	No	Music Artist, Rhapsody Channel Artist, Music Video Artist
Audiobook Title	3	0x03	No	Audio Book Title
Audiobook Author	4	0x04	No	Audio Book Artist
TV Episode	5	0x05	No	Audio Podcast Episode, TV Show Episode, Video Podcast Episode

Audio Podcast Name	6	0x06	No	Audio Podcast Name, Video Podcast Name
Movie Title	7	0x07	No	Movie Title
TV Title	8	0x08	No	TV Title
Music Video Title	9	0x09	No	Music Video Title
Non Indexing Source	10	0x0A	Yes	Non Indexing Source, only used in MetadataIcon_1
A/V In	11	0x0B	Yes	A/V In, only used in MetadataIcon_1
Bluetooth	12	0x0C	Yes	Non Indexing Bluetooth Audio Device, only used in MetadataIcon_1
Line In	13	0x0D	Yes	Line In, only used in MetadataIcon_1
Reserved	14-22	0x0E … 0x16	No	Reserved for any additional sources or Metadata Icons.
Blank	23	0x17	No	No Icon
Unknown	24	0x18	No	Unknown Icon
Reserved	25-255	0x19 … 0xFF	No	Reserved for future use.

2.10.1.19 LBP-REQ-019753/B-Source Icons (TcSE ROIN-301569-1)

Icon Name	**Source Icon ID**
Invalid	0x00
AM	0x01
CD	0x02
Sirius	0x03
Audio Video In	0x04
User device - USB	0x05
User device - phone	0x06
User device - media player	0x07
User device - SD Card	0x08
User device - WiFi	0x09
User Device- Bluetooth Audio	0x0A
Line In	0x0B
FM	0x0C
DAB	0x0D
Reserved	0x0E…0xFD
Blank	0xFE
Unknown Source	0xFF

2.10.1.20 LBP-REQ-019754/A-Phone Icons (TcSE ROIN-301570-1)

Icon Name	**Call/Phone Type Icon**
Invalid	0x00
general (other) phone	0x01
home	0x02
mobile	0x03
office	0x04
incoming	0x05
outgoing	0x06
missed	0x07

Reserved	0x08…0xFD
Blank	0xFE
Unknown	0xFF

2.10.1.21 LBP-REQ-019755/A-Navigation Icons (TcSE ROIN-301571-1)

Icon Name	Direction
Invalid	0x00
Crow's Flight N	0x01
Crow's Flight NE	0x02
Crow's Flight E	0x03
Crow's Flight SE	0x04
Crow's Flight S	0x05
Crow's Flight SW	0x06
Crow's Flight W	0x07
Crow's Flight NW	0x08
Reserved	0x09..0xFD
Blank	0xFE
Unknown	0xFF

2.10.1.22 LBP-REQ-019756/C-List Server Generic Media 1 - Media List Structure (TcSE ROIN-301572-1)

Media Root List						
ListServerID =	Generic Media1 (0x02)					
ActiveListID =	Media Root (0x0000)					
ParentList ID =	Media Root (0x0000)					
NbrItemsInSelection=	20					
ItemIndex	***DataType***	***ActivationEvent***	***Object Type***	***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	{Media Root List}	Invalid	Invalid
0x0001	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{Source Icon} {Source #:Source Name}	Goto Active List ID 0x0001	Set Source to Entry 1
0x0002	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{Source Icon} {Source #:Source Name}	Goto Active List ID 0x0001	Set Source to Entry 2
0x0003	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{Source Icon} {Source #:Source Name}	Goto Active List ID 0x0001	Set Source to Entry 3
0x0004	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{Source Icon} {Source #:Source Name}	Goto Active List ID 0x0001	Set Source to Entry 4
				…		
0x0014	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{Source Icon} {Source #:Source Name}	Invalid	Set Source to Entry 20

Non Indexing Device
can only use setItem

Example 1:

0x0001	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{USB Icon} 1:USB	Goto Active List ID 0x0001	Set Source to Entry 1
0x0002	Media Type (0x20)	Supported (0x1)	List Object (0x2)	{USB Icon} 2:USB	Goto Active List ID 0x0001	Set Source to Entry 2

Example 2:

0x0001	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x07} 1:Pandora	Invalid	Set Source to Entry 1
0x0002	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x07} 2:iHeart Radio	Invalid	Set Source to Entry 2
0x0003	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x07} 3:CarPlay	Invalid	Set Source to Entry 3
0x0004	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x0A} 4:BTAudio	Invalid	Set Source to Entry 4

Example: Now Playing (Applink Source selected)

Source Info: {device name}
Metadata1: {mainField 1}
Metadata2: {mainField 2}
Metadata Icon 1 and 2: 0x17 (No Icon)

Note: Fields described in examples are referenced in S28 63.4.2.41 TP-LOG-GTPC-160692-2-SID-79-MediaInformation_St

Indexing Device Features						
ListServerID =	Generic Media1 (0x02)					
ActiveListID =	Indexable Features (0x0001)					
ParentList ID =	Media Root (0x0000)					
NbrItemsInSelection=	4					
ItemIndex	***DataType***	***ActivationEvent***	***Object Type***	***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	{Device Name}	Invalid	Invalid
0x0001	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	Now Playing	Invalid	Play Entry 1
0x0002	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	Play All	Invalid	Play Entry 2
0x0003	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	Shuffle All	Invalid	Play Entry 3

Example:

0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Ken's iPod	Invalid	Invalid

2.10.1.23 LBP-REQ-019757/B-List Server Phone Info - Phone List Structure (TcSE ROIN-301575-1)

Note: This list order cannot change. There is a dependancy on the cluster for which list item number to capture.

Phone Root List						
ListServerID =	Phone Info (0x03)					
ActiveListID =	Phone Root (0x0000)					
ParentList ID =	Phone Root (0x0000)					
NbrItemsInSelection =	5					
ItemIndex	***DataType***	***ActivationEvent***	***Object Type***	***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Phone	Invalid	Invalid
0x0001	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	Quick Dial	Goto ActiveListID 0x0001 or 0x0006	Invalid
0x0002	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	All Calls	Goto ActiveListID 0x0002 or 0x0006	Invalid
0x0003	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	Incoming Calls	Goto ActiveListID 0x0003 or 0x0006	Invalid
0x0004	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	Outgoing Calls	Goto ActiveListID 0x0004 or 0x0006	Invalid
0x0005	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	Missed Calls	Goto ActiveListID 0x0005 or 0x0006	Invalid

Quick Dial						
ListServerID =	Phone Info (0x03)					
ActiveListID =	Quick Dial (0x0001)					
ParentList ID =	Phone Root (0x0000)					
NbrItemsInSelection =	9					
ItemIndex	***DataType***	***ActivationEvent***	***Object Type***	***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Quick Dial	Invalid	Invalid
0x0001	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{null}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 1
0x0002	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{null}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 2
0x0003	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{null}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 3
0x0004	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{null}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 4
0x0005	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{null}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 5
0x0006	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{null}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 6
0x0007	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{null}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 7
0x0008	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{null}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 8
0x0009	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{null}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 9

All Calls						
ListServerID =	Phone Info (0x03)					
ActiveListID =	All Calls (0x0002)					
ParentList ID =	Phone Root(0x0000)					
NbrItemsInSelection =	25					
ItemIndex	***DataType***	***ActivationEvent***		***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	All Calls	Invalid	Invalid
0x0001	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 1
0x0002	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 2
0x0003	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 3
0x0004	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 4
0x0005	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 5
0x0006	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 6
0x0007	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 7
0x0008	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 8
0x0009	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 9
0x000A	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 10
...						
0x0019	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 25

Incoming Calls						
ListServerID =	Phone Info (0x03)					
ActiveListID =	Incoming Calls (0x0003)					
ParentList ID =	Phone Root (0x0000)					
NbrItemsInSelection =	25					
ItemIndex	***DataType***	***ActivationEvent***		***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Incoming Calls	Invalid	Invalid
0x0001	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 1
0x0002	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 2
0x0003	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 3
0x0004	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 4
0x0005	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 5
0x0006	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 6
0x0007	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 7
0x0008	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 8
0x0009	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 9
0x000A	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 10
		...				
0x0019	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 25

Outgoing Calls						
ListServerID =	Phone Info (0x03)					
ActiveListID =	Outgoing Calls (0x0004)					
ParentList ID =	Phone Root(0x0000)					
NbrItemsInSelection =	25					
ItemIndex	***DataType***	***ActivationEvent***		***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x2)	Not Supported (0x0)	List Label	Outgoing Calls	Invalid	Invalid
0x0001	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 1
0x0002	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 2
0x0003	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 3
0x0004	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 4
0x0005	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 5
0x0006	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 6
0x0007	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 7
0x0008	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 8
0x0009	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 9
0x000A	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 10
		...				
0x0019	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 25

MIssed Calls						
ListServerID =	Phone Info (0x03)					
ActiveListID =	Missed Calls (0x0005)					
ParentList ID =	Phone Root (0x0000)					
NbrItemsInSelection=	25					
ItemIndex	***DataType***	***ActivationEvent***		***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Missed Calls	Invalid	Invalid
0x0001	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 1
0x0002	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 2
0x0003	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 3
0x0004	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 4
0x0005	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 5
0x0006	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 6
0x0007	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 7
0x0008	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 8
0x0009	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 9
0x000A	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 10
		...				
0x0019	Phone CallerID (0x60)	Supported (0x1)	Entry Object (0x1)	{CallTypeIcon}{PhoneTypeIcon}{CallerID}	Invalid	Place Call to entry 25

Empty Lists						
ListServerID =	Phone Info (0x03)					
ActiveListID =	History Empty(0x0006)					
ParentList ID =	Phone Root (0x0000)					
NbrItemsInSelection=	1					
ItemIndex	***DataType***	***ActivationEvent***	***Object Type***	***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	No Phonebook	Invalid	Invalid
0x0001	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	Return	Goto Parent	Invalid

2.10.1.24 LBP-REQ-019758/B-List Server Navigation Info - Navigation List Structure (TcSE ROIN-301576-1)

Navigation Root List

ListServerID =	Navigation Info (0x01)
ActiveListID =	Navigation Root (0x0000)
ParentList ID =	Navigation Root (0x0000)
NbrItemsInSelection =	4

ItemIndex	DataType	ActivationEvent	Object Type	ItemDescriptor	getItem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Quick Navigation	Invalid	Invalid
0x0001	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	My Home	Invalid	Set Dest to entry 1 (If MyHome Not Saved, Goto Active List ID 0x0004)
0x0002	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	Previous Destinations	Goto Active List ID 0x0003 or 0x0004	Invalid
0x0003	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	Favorites	Goto Active List ID 0x0001 or 0x0004	Invalid
0x0004	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	POI Nearby	Goto Active List ID 0x0002	Invalid

Favorites

ListServerID =	Navigation Info (0x01)
ActiveListID =	Favorites (0x0001)
ParentList ID =	Navigation Root (0x0000)
NbrItemsInSelection =	25

ItemIndex	DataType	ActivationEvent	Object Type	ItemDescriptor	getItem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Favorites	Invalid	Invalid
0x0001	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest1}	Invalid	Set Dest to entry 1
0x0002	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest2}	Invalid	Set Dest to entry 2
0x0003	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest3}	Invalid	Set Dest to entry 3
0x0004	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest4}	Invalid	Set Dest to entry 4
0x0005	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest5}	Invalid	Set Dest to entry 5
0x0006	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest6}	Invalid	Set Dest to entry 6
0x0007	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest7}	Invalid	Set Dest to entry 7
0x0008	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest8}	Invalid	Set Dest to entry 8
0x0009	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest9}	Invalid	Set Dest to entry 9
0x000A	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest10}	Invalid	Set Dest to entry 10
			…			
0x0019	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{FavDest25}	Invalid	Set Dest to entry 25

POI Nearby

ListServerID =	Navigation Info (0x01)
ActiveListID =	POI Nearby (0x0002)
ParentList ID =	Navigation Root (0x0000)
NbrItemsInSelection =	5

ItemIndex	DataType	ActivationEvent	Object Type	ItemDescriptor	getItem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	POI Nearby	Invalid	Invalid
0x0001	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	{Favorite_POI_Cat_1}	Goto ActiveListID 0x0004 or 0x0005	Invalid
0x0002	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	{Favorite_POI_Cat_2}	Goto ActiveListID 0x0004 or 0x0005	Invalid
0x0003	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	{Favorite_POI_Cat_3}	Goto ActiveListID 0x0004 or 0x0005	Invalid
0x0004	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	{Favorite_POI_Cat_4}	Goto ActiveListID 0x0004 or 0x0005	Invalid
0x0005	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	{Favorite_POI_Cat_5}	Goto ActiveListID 0x0004 or 0x0005	Invalid

Previous Destinations

ListServerID =	Navigation Info (0x01)
ActiveListID =	Previous Destinations (0x0003)
ParentList ID =	Navigation Root (0x0000)
NbrItemsInSelection=	20

ItemIndex	DataType	ActivationEvent	Object Type	ItemDescriptor	getItem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Previous Destinations	Invalid	Invalid
0x0001	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_1}	Invalid	Set Dest to entry 1
0x0002	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_2}	Invalid	Set Dest to entry 2
0x0003	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_3}	Invalid	Set Dest to entry 3
0x0004	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_4}	Invalid	Set Dest to entry 4
0x0005	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_5}	Invalid	Set Dest to entry 5
0x0006	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_6}	Invalid	Set Dest to entry 6
0x0007	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_7}	Invalid	Set Dest to entry 7
0x0008	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_8}	Invalid	Set Dest to entry 8
0x0009	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_9}	Invalid	Set Dest to entry 9
0x000A	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_10}	Invalid	Set Dest to entry 10
		…				
0x0014	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	{Prev_Dest_20}	Invalid	Set Dest to entry 20

No Matches Found

ListServerID =	Navigation Info (0x01)
ActiveListID =	No Matches Found (0x0004)
ParentList ID =	POI Nearby (0x0002) or Navigation Root List (0x0000)
NbrItemsInSelection=	1

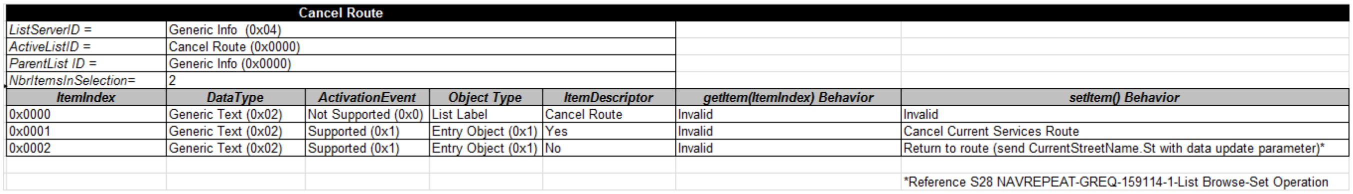
ItemIndex	DataType	ActivationEvent	Object Type	ItemDescriptor	getItem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	No Matches Found	Invalid	Invalid
0x0001	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	Return	Goto Parent	Invalid

POI List

ListServerID =	Navigation Info (0x01)
ActiveListID =	POI List(0x0005)
ParentList ID =	POI Nearby (0x0002)
NbrItemsInSelection=	20

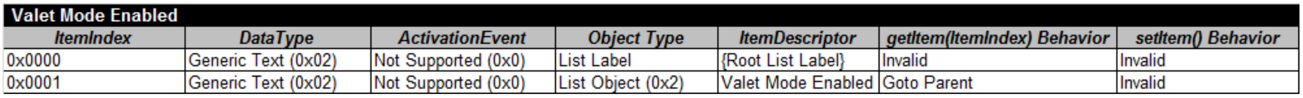
ItemIndex	DataType	ActivationEvent	Object Type	ItemDescriptor	getItem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	{POI Type}	Invalid	Invalid
0x0001	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 1
0x0002	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 2
0x0003	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 3
0x0004	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 4
0x0005	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 5
0x0006	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 6
0x0007	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 7
0x0008	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 8
0x0009	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 9
0x000A	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 10
		…				
0x0014	Navigation POI (0x40)	Supported (0x1)	Entry Object (0x1)	{Direction}{Distance}{POI Name}	Invalid	Set Dest to entry 20

2.10.1.25 LBP-REQ-019759/A-List Server Generic List 1 - Sync Services List Structure (TcSE ROIN-301589-1)



Cancel Route						
ListServerID =	Generic Info (0x04)					
ActiveListID =	Cancel Route (0x0000)					
ParentList ID =	Generic Info (0x0000)					
NbrItemsInSelection=	2					
ItemIndex	DataType	ActivationEvent	Object Type	ItemDescriptor	getItem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Cancel Route	Invalid	Invalid
0x0001	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	Yes	Invalid	Cancel Current Services Route
0x0002	Generic Text (0x02)	Supported (0x1)	Entry Object (0x1)	No	Invalid	Return to route (send CurrentStreetName.St with data update parameter)*
						*Reference S28 NAVREPEAT-GREQ-159114-1-List Browse-Set Operation

2.10.1.26 LBP-REQ-019760/A-List Server Generic List 1 - Valet Mode List Structure (TcSE ROIN-301590-1)



Valet Mode Enabled						
ItemIndex	DataType	ActivationEvent	Object Type	ItemDescriptor	getItem(ItemIndex) Behavior	setItem() Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	{Root List Label}	Invalid	Invalid
0x0001	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	Valet Mode Enabled	Goto Parent	Invalid

2.10.1.27 LBP-REQ-089676/B-LBP Media definition for GAL/CarPlay.

When a projection session is active (e.g. Apple CarPlay, Android Auto/GAL, Baidu CarLife), the APIM shall send source information in the list browse protocol response. If other APIM internal media sources are still available during the projection session, the APIM shall provide these items in the list browse protocol response.

	LBP List Structure					
	DataType	**ActivationEvent**	**Object Type**	**ItemDescriptor**	**getItem(ItemIndex) Behavior**	**setItem() Behavior**
Apple CarPlay	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x07} {Source #: Device Name}	Invalid	Set Source
Android Auto / GAL	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x07} {Source #: Android Auto}	Invalid	Set Source
Baidu CarLife	Media Type (0x20)	Supported (0x1)	Entry Object (0x1)	{0x07} {Source #: CarLife Music}	Invalid	Set Source

When a projection session is active and projection media is active, selecting the projection source from the list browse will direct the system to now playing cluster screen. Examples:

Apple CarPlay	**Android Auto**	**Baidu CarLife**
Source Info: {Device Name}	Source Info: “Android Auto”	Source Info: “CarLife”
Metadata1:{MediaItemArtist}	Metadata1: {Artist}	Metadata1: {Artist}
Metadata2: {MediaItemTitle}	Metadata2: {Song}	Metadata2: {Song}
Metadata Icon 1 and 2: 0x17 (no icon)	Metadata Icon 1 and 2: 0x17 (no icon)	Metadata Icon 1 and 2: 0x17 (no icon)

2.10.1.28 LBP-REQ-089678/B-LBP Phone definition for GAL/CarPlay.

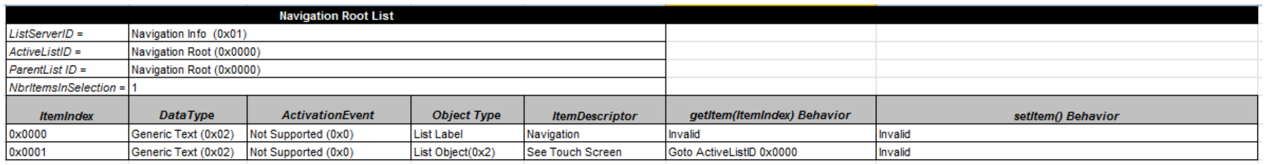
When GAL is active, the APIM shall send GAL phone information in the list browse protocol response. The APIM shall follow the currently define phone list structure.

When CarPlay is active, the following table shall be transferred via list browse protocol.

Phone Root List CarPlay						
ListServerID =	Phone Info (0x03)					
ActiveListID =	Phone Root (0x0000)					
ParentList ID =	Phone Root (0x0000)					
NbrItemsInSelection =	1					
ItemIndex	***DataType***	***ActivationEvent***	***Object Type***	***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Phone	Invalid	Invalid
0x0001	Generic Text (0x02)	Not Supported (0x0)	List Object (0x2)	See Touch Screen	Goto ActiveListID 0x0000	Invalid

2.10.1.29 LBP-REQ-089684/A-LBP Navigation definition for GAL/CarPlay

When GAL/CarPlay is active, the APIM shall send “See touch screen” for item one in the navigation root list in list browse protocol response. There shall no other items in the response.

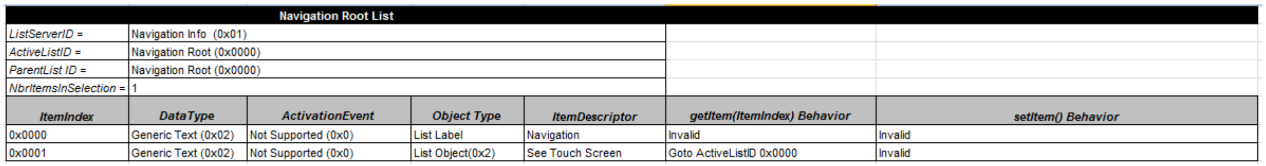


Navigation Root List						
ListServerID =	Navigation Info (0x01)					
ActiveListID =	Navigation Root (0x0000)					
ParentList ID =	Navigation Root (0x0000)					
NbrItemsInSelection =	1					
ItemIndex	***DataType***	***ActivationEvent***	***Object Type***	***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Navigation	Invalid	Invalid
0x0001	Generic Text (0x02)	Not Supported (0x0)	List Object(0x2)	See Touch Screen	Goto ActiveListID 0x0000	Invalid

When GAL/CarPlay is active, the APIM shall utilize the current street name statues to TP method to send “see touch screen”

2.10.1.30 LBP-REQ-222052/A-LBP Navigation definition for Mobile Navigation

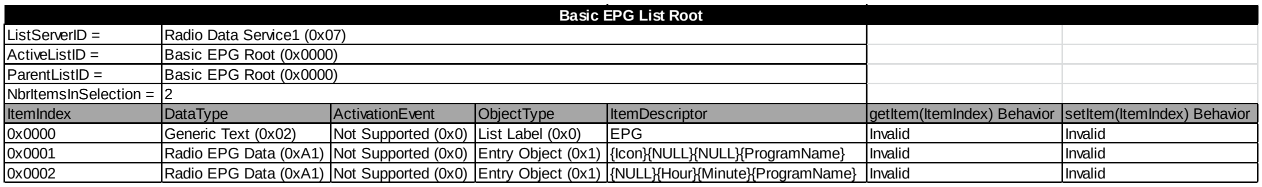
When Mobile Navigation is active, the APIM shall send “See touch screen” for item one in the navigation root list in list browse protocol response. There shall be no other items in the response.



Navigation Root List						
ListServerID =	Navigation Info (0x01)					
ActiveListID =	Navigation Root (0x0000)					
ParentList ID =	Navigation Root (0x0000)					
NbrItemsInSelection =	1					
ItemIndex	***DataType***	***ActivationEvent***	***Object Type***	***ItemDescriptor***	***getItem(ItemIndex) Behavior***	***setItem() Behavior***
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label	Navigation	Invalid	Invalid
0x0001	Generic Text (0x02)	Not Supported (0x0)	List Object(0x2)	See Touch Screen	Goto ActiveListID 0x0000	Invalid

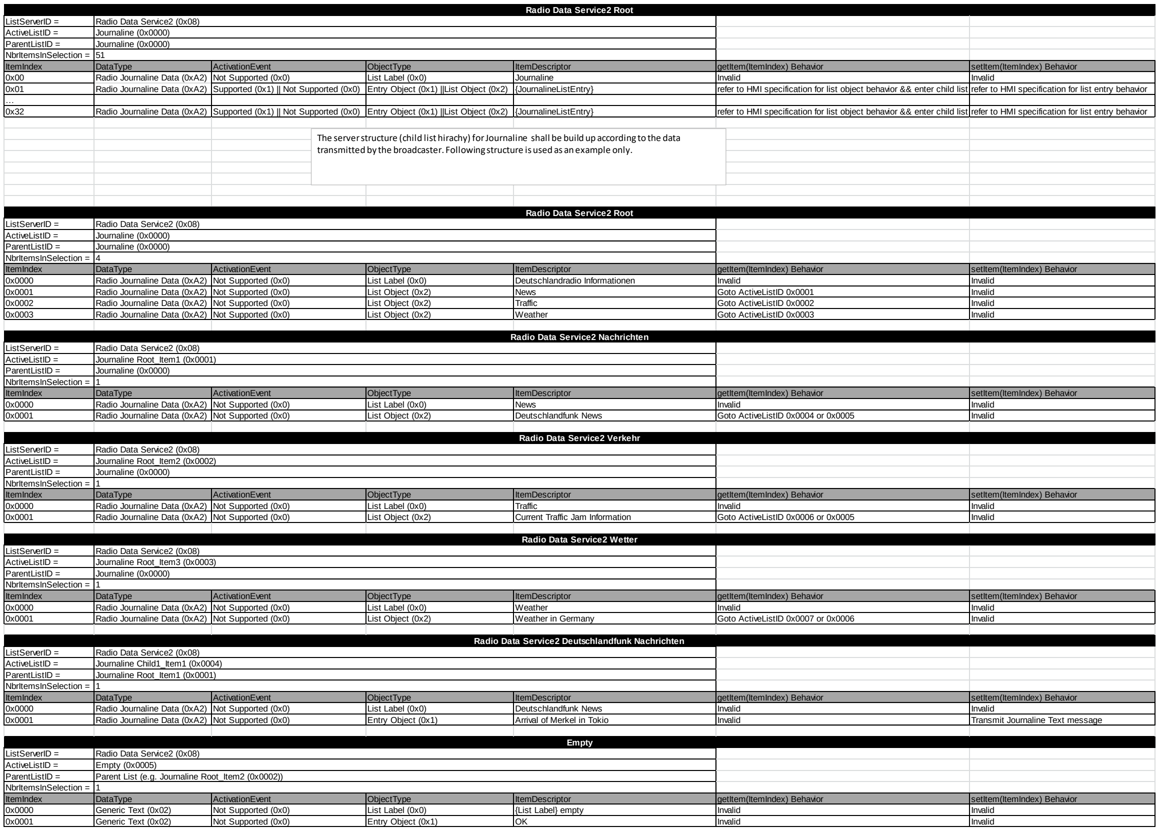
When Mobile Navigation is active, the APIM shall utilize the current street name statues to TP method to send “see touch screen”

2.10.1.31 LBP-REQ-132690/B-List Server RadioDataService1 - EPG List Structure for Centerstack HMI



Basic EPG List Root						
ListServerID =	Radio Data Service1 (0x07)					
ActiveListID =	Basic EPG Root (0x0000)					
ParentListID =	Basic EPG Root (0x0000)					
NbrItemsInSelection =	2					
ItemIndex	DataType	ActivationEvent	ObjectType	ItemDescriptor	getItem(ItemIndex) Behavior	setItem(ItemIndex) Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label (0x0)	EPG	Invalid	Invalid
0x0001	Radio EPG Data (0xA1)	Not Supported (0x0)	Entry Object (0x1)	{Icon}{NULL}{NULL}{ProgramName}	Invalid	Invalid
0x0002	Radio EPG Data (0xA1)	Not Supported (0x0)	Entry Object (0x1)	{NULL}{Hour}{Minute}{ProgramName}	Invalid	Invalid

2.10.1.32 LBP-REQ-132691/B-List Server RadioDataService2 - Journaline List Structure for Centerstack HMI



Radio Data Service2 Root						
ListServerID =	Radio Data Service2 (0x08)					
ActiveListID =	Journaline (0x0000)					
ParentListID =	Journaline (0x0000)					
NbrItemsInSelection =	51					
ItemIndex	DataType	ActivationEvent	ObjectType	ItemDescriptor	getItem(ItemIndex) Behavior	setItem(ItemIndex) Behavior
0x00	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Label (0x0)	Journaline	Invalid	Invalid
0x01	Radio Journaline Data (0xA2)	Supported (0x1) || Not Supported (0x0)	Entry Object (0x1) ||List Object (0x2)	{JournalineListEntry}	refer to HMI specification for list object behavior && enter child list	refer to HMI specification for list entry behavior
…						
0x32	Radio Journaline Data (0xA2)	Supported (0x1) || Not Supported (0x0)	Entry Object (0x1) ||List Object (0x2)	{JournalineListEntry}	refer to HMI specification for list object behavior && enter child list	refer to HMI specification for list entry behavior

The server structure (child list hirachy) for Journaline shall be build up according to the data transmitted by the broadcaster. Following structure is used as an example only.

Radio Data Service2 Root						
ListServerID =	Radio Data Service2 (0x08)					
ActiveListID =	Journaline (0x0000)					
ParentListID =	Journaline (0x0000)					
NbrItemsInSelection =	4					
ItemIndex	DataType	ActivationEvent	ObjectType	ItemDescriptor	getItem(ItemIndex) Behavior	setItem(ItemIndex) Behavior
0x0000	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Label (0x0)	Deutschlandradio Informationen	Invalid	Invalid
0x0001	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Object (0x2)	News	Goto ActiveListID 0x0001	Invalid
0x0002	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Object (0x2)	Traffic	Goto ActiveListID 0x0002	Invalid
0x0003	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Object (0x2)	Weather	Goto ActiveListID 0x0003	Invalid

Radio Data Service2 Nachrichten						
ListServerID =	Radio Data Service2 (0x08)					
ActiveListID =	Journaline Root_Item1 (0x0001)					
ParentListID =	Journaline (0x0000)					
NbrItemsInSelection =	1					
ItemIndex	DataType	ActivationEvent	ObjectType	ItemDescriptor	getItem(ItemIndex) Behavior	setItem(ItemIndex) Behavior
0x0000	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Label (0x0)	News	Invalid	Invalid
0x0001	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Object (0x2)	Deutschlandfunk News	Goto ActiveListID 0x0004 or 0x0005	Invalid

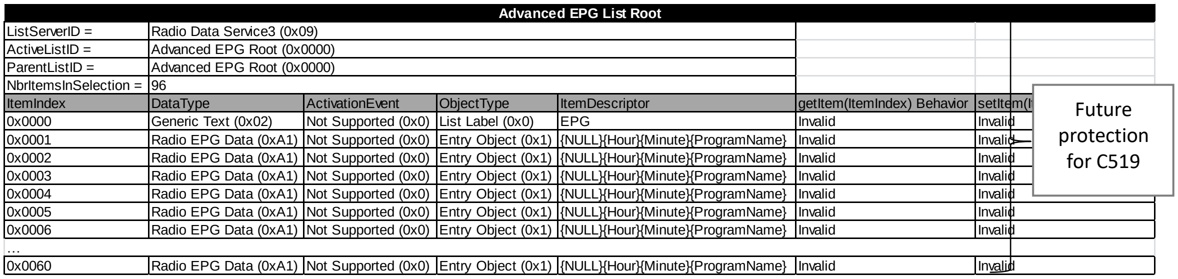
Radio Data Service2 Verkehr						
ListServerID =	Radio Data Service2 (0x08)					
ActiveListID =	Journaline Root_Item2 (0x0002)					
ParentListID =	Journaline (0x0000)					
NbrItemsInSelection =	1					
ItemIndex	DataType	ActivationEvent	ObjectType	ItemDescriptor	getItem(ItemIndex) Behavior	setItem(ItemIndex) Behavior
0x0000	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Label (0x0)	Traffic	Invalid	Invalid
0x0001	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Object (0x2)	Current Traffic Jam Information	Goto ActiveListID 0x0006 or 0x0005	Invalid

Radio Data Service2 Wetter						
ListServerID =	Radio Data Service2 (0x08)					
ActiveListID =	Journaline Root_Item3 (0x0003)					
ParentListID =	Journaline (0x0000)					
NbrItemsInSelection =	1					
ItemIndex	DataType	ActivationEvent	ObjectType	ItemDescriptor	getItem(ItemIndex) Behavior	setItem(ItemIndex) Behavior
0x0000	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Label (0x0)	Weather	Invalid	Invalid
0x0001	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Object (0x2)	Weather in Germany	Goto ActiveListID 0x0007 or 0x0006	Invalid

Radio Data Service2 Deutschlandfunk Nachrichten						
ListServerID =	Radio Data Service2 (0x08)					
ActiveListID =	Journaline Child1_Item1 (0x0004)					
ParentListID =	Journaline Root_Item1 (0x0001)					
NbrItemsInSelection =	1					
ItemIndex	DataType	ActivationEvent	ObjectType	ItemDescriptor	getItem(ItemIndex) Behavior	setItem(ItemIndex) Behavior
0x0000	Radio Journaline Data (0xA2)	Not Supported (0x0)	List Label (0x0)	Deutschlandfunk News	Invalid	Invalid
0x0001	Radio Journaline Data (0xA2)	Not Supported (0x0)	Entry Object (0x1)	Arrival of Merkel in Tokio	Invalid	Transmit Journaline Text message

Empty						
ListServerID =	Radio Data Service2 (0x08)					
ActiveListID =	Empty (0x0005)					
ParentListID =	Parent List (e.g. Journaline Root_Item2 (0x0002))					
NbrItemsInSelection =	1					
ItemIndex	DataType	ActivationEvent	ObjectType	ItemDescriptor	getItem(ItemIndex) Behavior	setItem(ItemIndex) Behavior
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label (0x0)	{List Label} empty	Invalid	Invalid
0x0001	Generic Text (0x02)	Not Supported (0x0)	Entry Object (0x1)	OK	Invalid	Invalid

2.10.1.33 LBP-REQ-192091/A-List Server RadioDataService3 - Advanced EPG List Structure for Centerstack HMI



Advanced EPG List Root						
ListServerID =	Radio Data Service3 (0x09)					
ActiveListID =	Advanced EPG Root (0x0000)					
ParentListID =	Advanced EPG Root (0x0000)					
NbrItemsInSelection =	96					
ItemIndex	DataType	ActivationEvent	ObjectType	ItemDescriptor	getItem(ItemIndex) Behavior	setItem(I
0x0000	Generic Text (0x02)	Not Supported (0x0)	List Label (0x0)	EPG	Invalid	Invalid
0x0001	Radio EPG Data (0xA1)	Not Supported (0x0)	Entry Object (0x1)	{NULL}{Hour}{Minute}{ProgramName}	Invalid	Invalid
0x0002	Radio EPG Data (0xA1)	Not Supported (0x0)	Entry Object (0x1)	{NULL}{Hour}{Minute}{ProgramName}	Invalid	Invalid
0x0003	Radio EPG Data (0xA1)	Not Supported (0x0)	Entry Object (0x1)	{NULL}{Hour}{Minute}{ProgramName}	Invalid	Invalid
0x0004	Radio EPG Data (0xA1)	Not Supported (0x0)	Entry Object (0x1)	{NULL}{Hour}{Minute}{ProgramName}	Invalid	Invalid
0x0005	Radio EPG Data (0xA1)	Not Supported (0x0)	Entry Object (0x1)	{NULL}{Hour}{Minute}{ProgramName}	Invalid	Invalid
0x0006	Radio EPG Data (0xA1)	Not Supported (0x0)	Entry Object (0x1)	{NULL}{Hour}{Minute}{ProgramName}	Invalid	Invalid
…						
0x0060	Radio EPG Data (0xA1)	Not Supported (0x0)	Entry Object (0x1)	{NULL}{Hour}{Minute}{ProgramName}	Invalid	Invalid

Future protection for C519

2.11 LBP-FUN-REQ-130789/A-Server list update

2.11.1 Use Cases

2.11.1.1 LBP-UC-REQ-129267/A-Indicate Server List Update

Actors	System
Pre-conditions	Infotainment Network is active.
Scenario Description	A server list gets updated with new data.
Post-conditions	Update of server list is indicated to the client.
List of Exception Use Cases	Server list is empty: If the server list is empty the list server will indicated the status to the client.
Interfaces	G-HMI, Vehicle System Interface

2.11.2 White Box View

2.11.2.1 White Box Scenarios

2.11.2.1.1 LBP-SD-REQ-130402/A-Indicate server list update

Scenarios

Normal Usage

A server list is getting updated.

Constraints

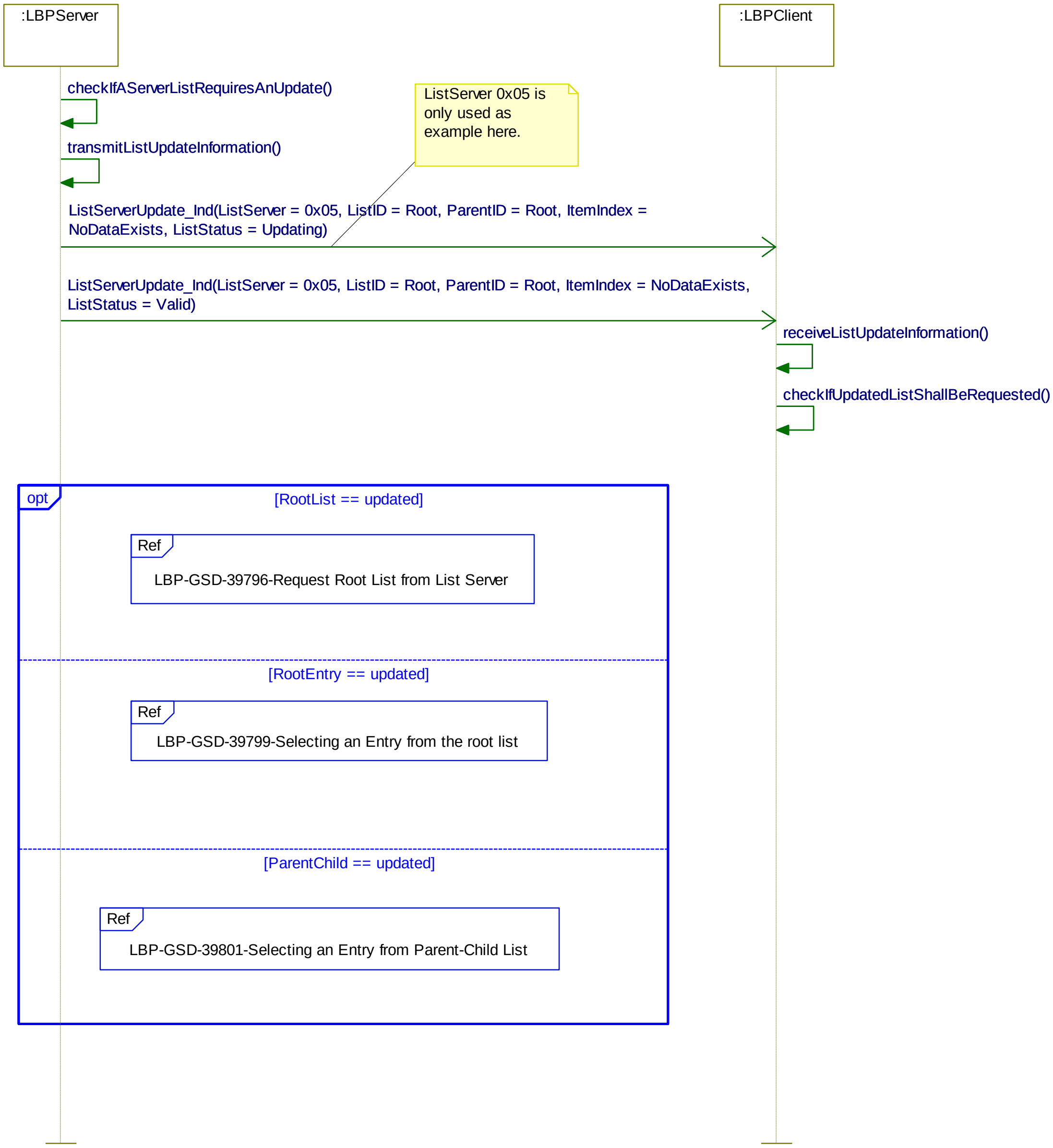
Pre-condition

Network is active.

Post-condition

Server list update information is indicated to the client.

Sequence Diagram


:LBPServer
:LBPClient
checkIfAServerListRequiresAnUpdate()
ListServer 0x05 is only used as example here.
transmitListUpdateInformation()
ListServerUpdate_Ind(ListServer = 0x05, ListID = Root, ParentID = Root, ItemIndex = NoDataExists, ListStatus = Updating)
ListServerUpdate_Ind(ListServer = 0x05, ListID = Root, ParentID = Root, ItemIndex = NoDataExists, ListStatus = Valid)
receiveListUpdateInformation()
checkIfUpdatedListShallBeRequested()
opt
[RootList == updated]
Ref
LBP-GSD-39796-Request Root List from List Server
[RootEntry == updated]
Ref
LBP-GSD-39799-Selecting an Entry from the root list
[ParentChild == updated]
Ref
LBP-GSD-39801-Selecting an Entry from Parent-Child List

3 Appendix: Reference Documents

Reference #	Document Title
1	
2	
3	
4	
5	