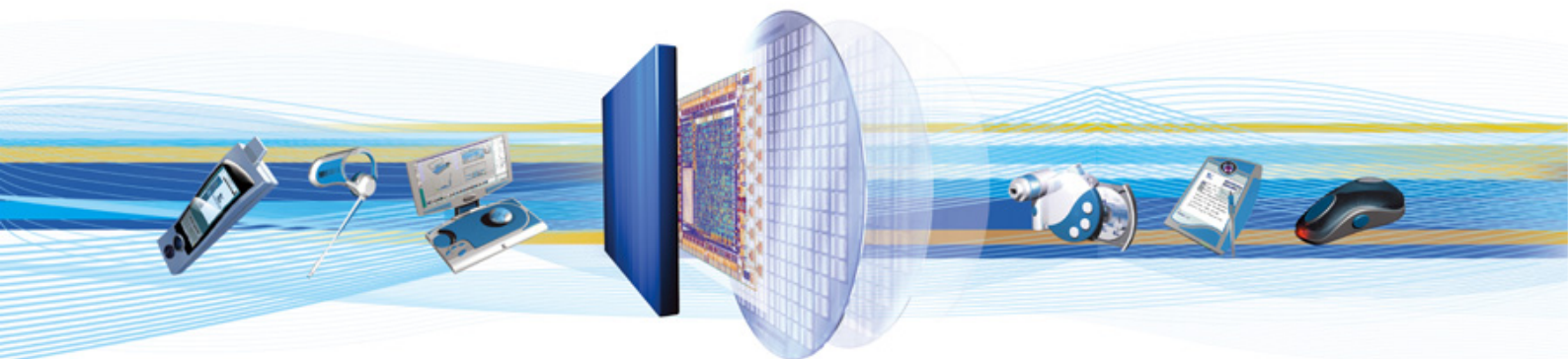




CSR Synergy Bluetooth 18.2.0

PHDC – Personal Health Device Communication Agent API Description

November 2011



Cambridge Silicon Radio Limited

Churchill House
Cambridge Business Park
Cowley Road
Cambridge CB4 0WZ
United Kingdom

Registered in England and Wales 3665875

Tel: +44 (0)1223 692000

Fax: +44 (0)1223 692001

www.csr.com



Contents

1	Introduction.....	5
1.1	Introduction and Scope	5
2	Description.....	6
2.1	Introduction.....	6
2.2	Sequence Overview	7
3	Interface Description.....	12
3.1	Configure.....	12
3.2	Device Configure	15
3.3	Device Capabilities	17
3.4	BT Link Establishment.....	19
3.5	Association.....	19
3.6	Disassociation	20
3.7	Suspend/Resume	22
3.8	Sending Data	24
3.9	Change of Measurement Unit.....	25
3.10	Setup BT Link	26
3.11	BT Link Release	26
4	Document References.....	28
5	Annex A: Usage of API's.....	32

List of Figures

Figure 1: Protocol Stack	6
Figure 2 : PHDC Agent Scenarios	11
Figure 3: Configuration Sequence	13
Figure 4: Device Configure Sequence	16
Figure 5: Device Search Sequence	18
Figure 6 Association Sequence	20
Figure 7: Disassociation Sequence.....	21
Figure 8: Suspend Sequence	22
Figure 9: Resume Sequence.....	23
Figure 10: Data send Sequence	25
Figure 11: Measurement Unit update Sequence	26
Figure 122 : Configuration and Data Transfer Sequence Example.....	33

List of Tables

Table 1 : Arguments for CsrBtPhdcAgConfigureEndpointReqSend	12
Table 2: Parameters in CSR_BT_PHDC_AG_CONFIGURE_ENDPOINT_IND	13
Table 3: Arguments for CsrBtPhdcAgConfigureEndPointResSend	14
Table 4: Parameters in a CSR_BT_PHDC_AG_CONFIGURE_ENDPOINT_CFM primitive	14
Table 5: Arguments for CsrBtPhdcAgConfigureMdsReqSend	15
Table 6: Arguments for CsrBtPhdcAgDimConfigureReqSend	16
Table 7: Parameters in a CSR_BT_PHDC_AG_DIM_OBJECT_IND primitive	17
Table 8: Arguments for CsrBtPhdcAgDimObjectResSend	17
Table 9: Parameters in a CSR_BT_PHDC_AG_DIM_ATTRIB_IND primitive.....	17
Table 10: Arguments for CsrBtPhdcAgDimAttribResSend	17
Table 11: Parameters in a CSR_BT_PHDC_AG_DIM_CONFIGURE_CFM primitive.....	17
Table 12: Arguments for CsrBtPhdcAgGetDeviceCapabReqSend	17
Table 13: Parameters in a CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_IND primitive.....	18
Table 14: Parameters in a CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_CFM primitive	19
Table 15: Parameters in CSR_BT_PHDC_AG_ACCEPT_BT_LINK_IND primitive	19
Table 16: Arguments for CsrBtPhdcAgAcceptBtLinkResSend	19
Table 17: Arguments for CsrBtPhdcAgAssociateReqSend	20
Table 18: Parameters in CSR_BT_PHDC_AG_ASSOCIATE_CFM primitive.....	20
Table 19: Arguments for CsrBtPhdcAgDisassociateReqSend function.....	21
Table 20: Parameters in a CSR_BT_PHDC_AG_DISASSOCIATE_CFM primitive	21
Table 21: Parameters in a CSR_BT_PHDC_AG_DISASSOCIATE_IND primitive.....	21
Table 22 : Argument to CsrBtPhdcAgSuspendReqSend function.....	22
Table 23: Parameters in a CSR_BT_PHDC_AG_SUSPEND_CFM primitive.....	22
Table 24: Parameters in a CSR_BT_PHDC_AG_SUSPEND_IND primitive	22
Table 25: Argument to CsrBtPhdcAgResumeReqSend function	23
Table 26: Parameters in a CSR_BT_PHDC_AG_RESUME_CFM primitive.....	23
Table 27: Parameters in a CSR_BT_PHDC_AG_RESUME_IND primitive	23
Table 28: Arguments for CsrBtPhdcAgDataReqSend function.....	24
Table 29: Members of structure CsrBtPhdcAgDataNode	24
Table 30: Members of structure CsrBtPhdcAgAbsoluteTime.....	25
Table 31: Parameters in a CSR_BT_PHDC_AG_DATA_CFM primitive	25

Table 32: Arguments of CsrBtPhdcAgChangeDimUnitCodeReqSend	25
Table 33: Parameters in a CSR_BT_PHDC_AG_CHANGE_DIM_UNIT_CODE_CFM primitive	26
Table 34: Parameters in CSR_BT_PHDC_AG_SETUP_BT_LINK_IND primitive.....	26
Table 35: Parameters in CSR_BT_PHDC_AG_BT_LINK_REL_IND primitive	27

1 Introduction

1.1 Introduction and Scope

ISO/IEEE 11073 standards enable communication between medical devices and external computer systems. IEEE 20601 defines a common framework for making an abstract model of personal health data available in transport independent transfer syntax required to establish logical connections between systems, provide presentation capabilities and services needed to perform communication tasks. This is implemented in CSR Synergy Bluetooth by Personal Health Device Communication (PHDC) application profile. PHDC application profile complies with Continua Design Guidelines version 1.0. Audience of the document are application developers for PHDC Agent. The interface provided in this document supports Weighing scale and Blood pressure monitor device specializations. Standard and Extended configurations are supported.

2 Description

2.1 Introduction

This document describes the interface provided by Personal Health Device Communication (PHDC) application profile. This profile is used for connecting application data source devices such as blood pressure monitors, weight scales, glucose meters, thermometers, and pulse oximeters to application data sink devices such as mobile phones, laptops, desktop computers, and health appliances without the need for cables.

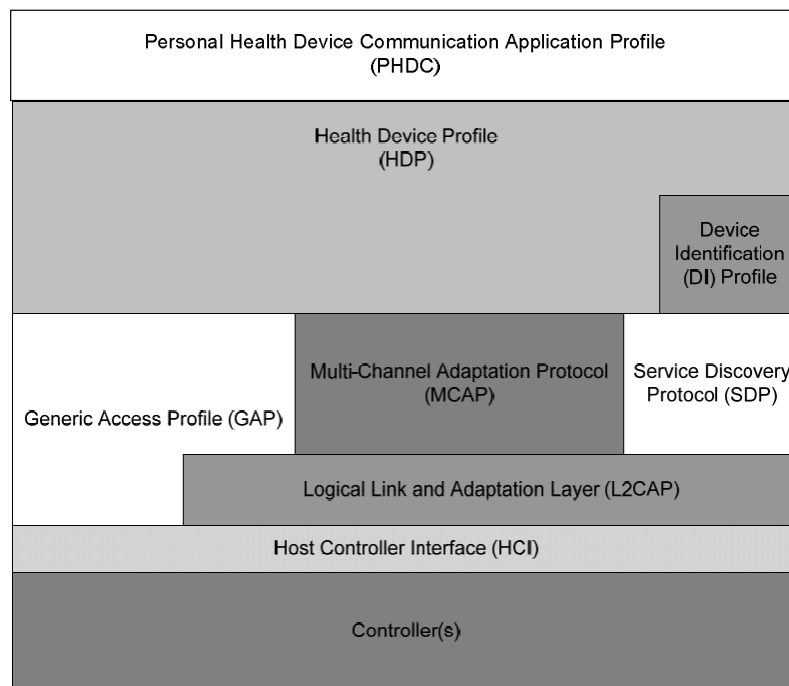
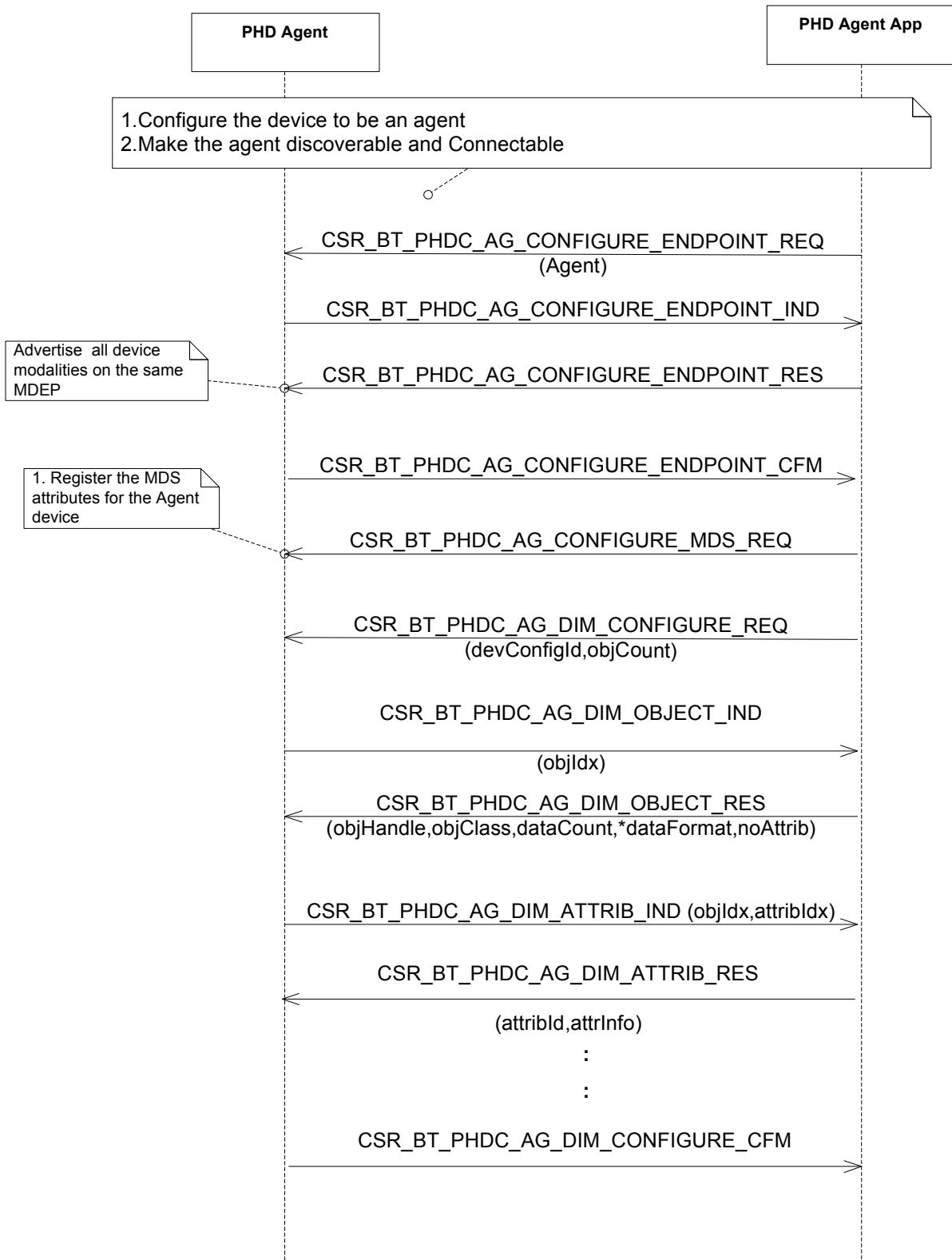
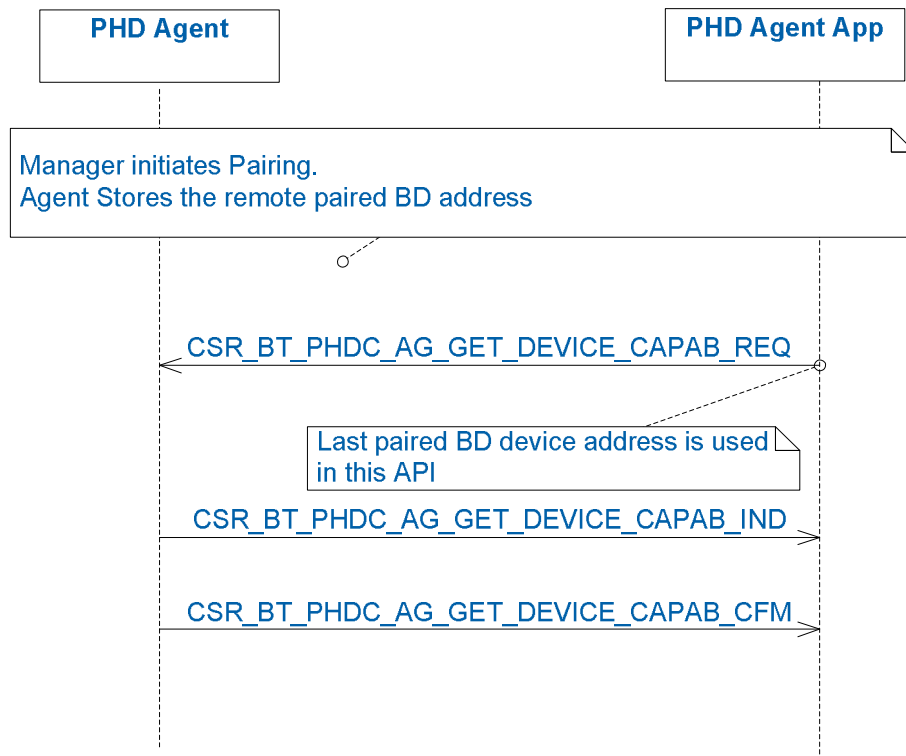


Figure 1: Protocol Stack

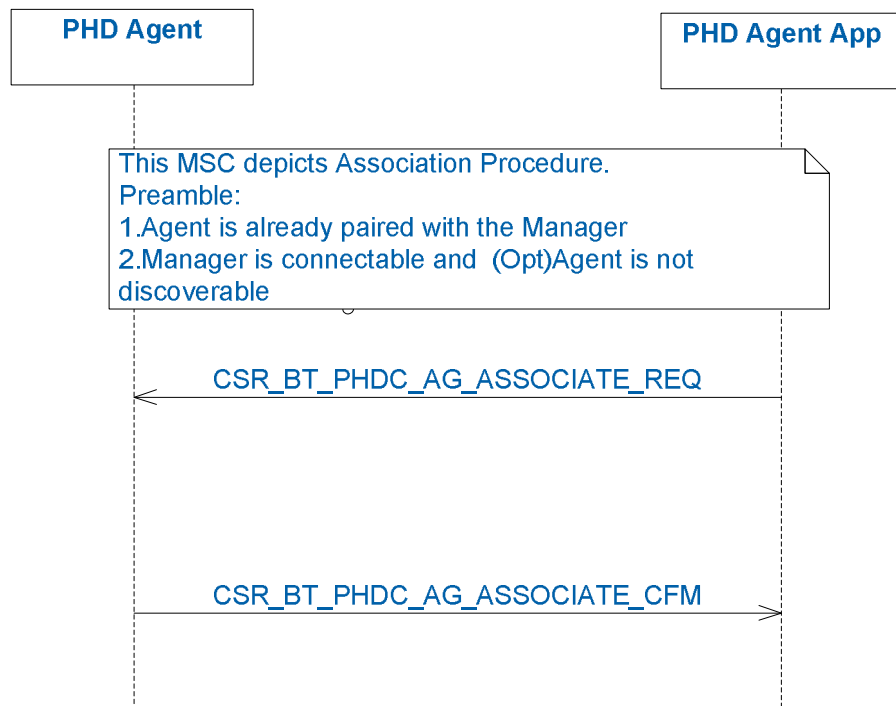
2.2 Sequence Overview



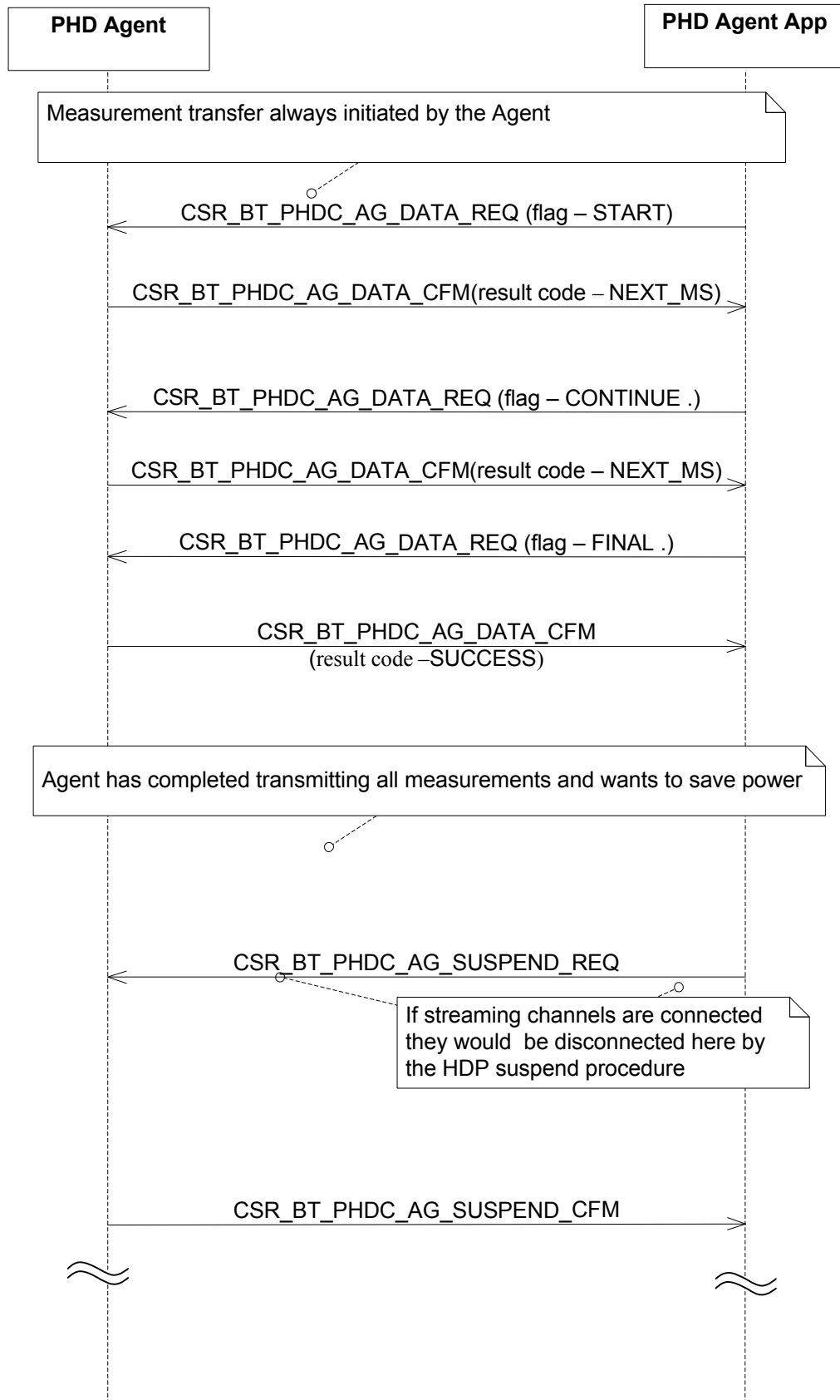
PHDC System Overview - 2



PHDC System Overview - 3



PHDC System Overview - 4



PHDC System Overview - 5

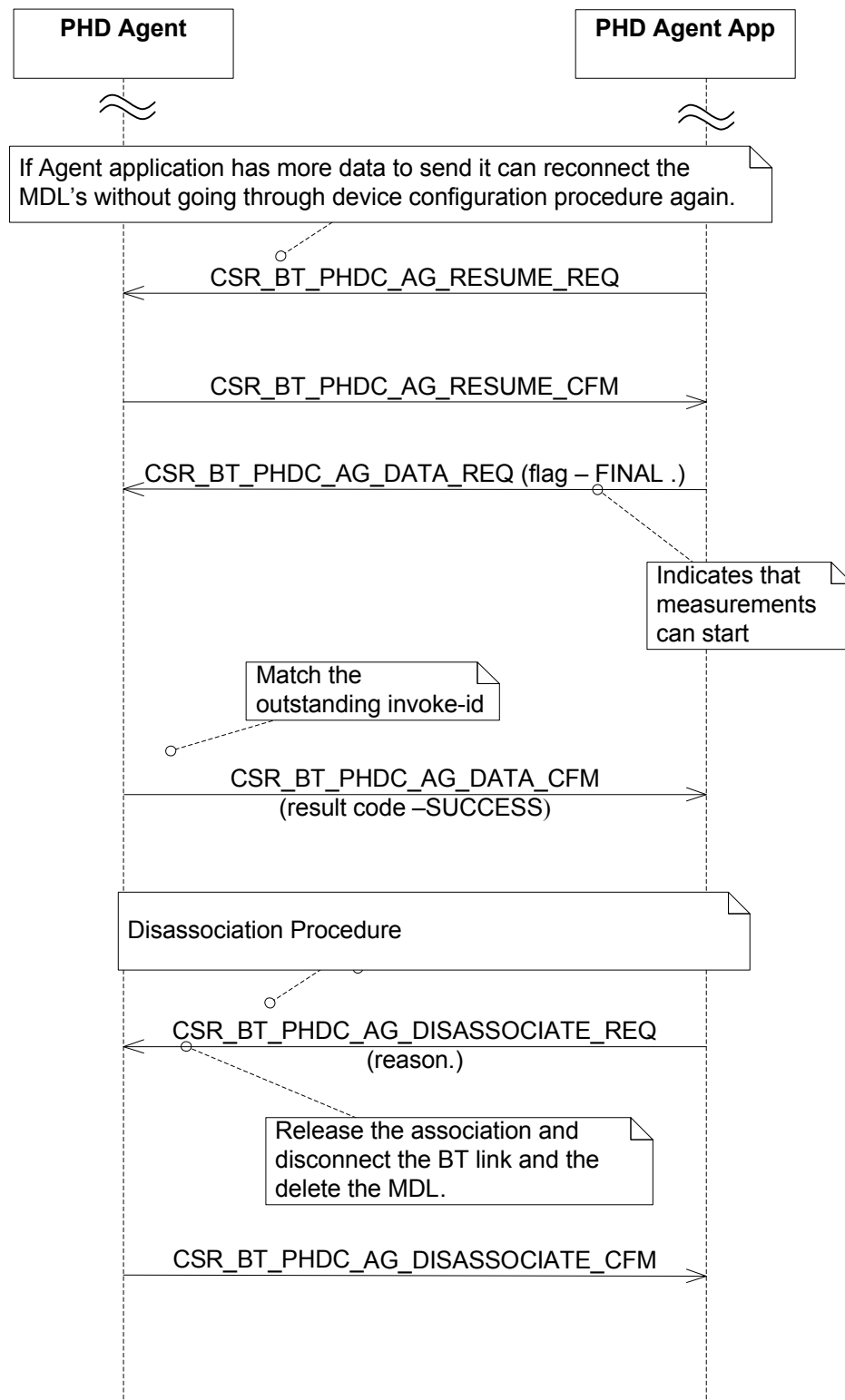


Figure 2 : PHDC Agent Scenarios

3 Interface Description

In this section, a series of MSCs are shown to explain the usage of the PHDC application profile. The primitives and the functions available to the application are also described in the subsections of this chapter.

3.1 Configure

Sending a CSR_BT_PHDC_AG_CONFIGURE_ENDPOINT_REQ registers a service record.

The Supported Features attribute for a HDP service record requires extra definition depending on how the application wants to advertise health device modalities. These device modalities can be provided on or more end-points or multiplexed on a single endpoint. These end points are identified by a MDEPID, Hence a phase of IND/RES transactions is initiated on receiving the REQ, to allow the profile to enumerate this information in the record.

On a successful configuration, a CSR_BT_PHDC_AG_CONFIGURE_ENDPOINT_CFM is sent to the application.

#	Type	Argument	Description
1	CsrSchedQid	qId	Queue handle of the application
2	dm_security_level_t	secLevel	Security level of the device being activated
3	CsrCharString*	serviceName	The ServiceName attribute is a string containing the name of the service represented by a service record.
4	CsrCharString*	serviceDescription	String containing a brief description of the service. Less than 200 characters.
5	CsrCharString*	providerName	String containing the name of the person or organization providing the service
6	CsrTime	sniffTimeOut	Time (in ms) the MCL must be idle before requesting sniff mode for the connection. If set to 0x0000, MCAP will never request sniff mode for the connection.
7	CsrUInt8	numOfMdep	Number of end points to advertise in the service record
8	CsrUInt8	supportedProcedures	<p>This is a one byte bit-mask that indicates the MCAP procedures that are to be supported by HDP service</p> <p>CSR_BT_HDP_SUPPORT_RECONNECT_INITIATION –</p> <p>Supports Reconnect Initiation 3</p> <p>CSR_BT_HDP_SUPPORT_RECONNECT_ACCEPTANCE –</p> <p>Supports Reconnect Acceptance 4</p> <p>CSR_BT_HDP_SUPPORT_CSP –</p> <p>Supports Clock Synchronization Protocol</p> <p>CSR_BT_HDP_SUPPORT_SYNC_MASTER_ROLE –</p> <p>Supports Sync-Master Role</p>

Table 1 : Arguments for CsrBtPhdcAgConfigureEndpointReqSend

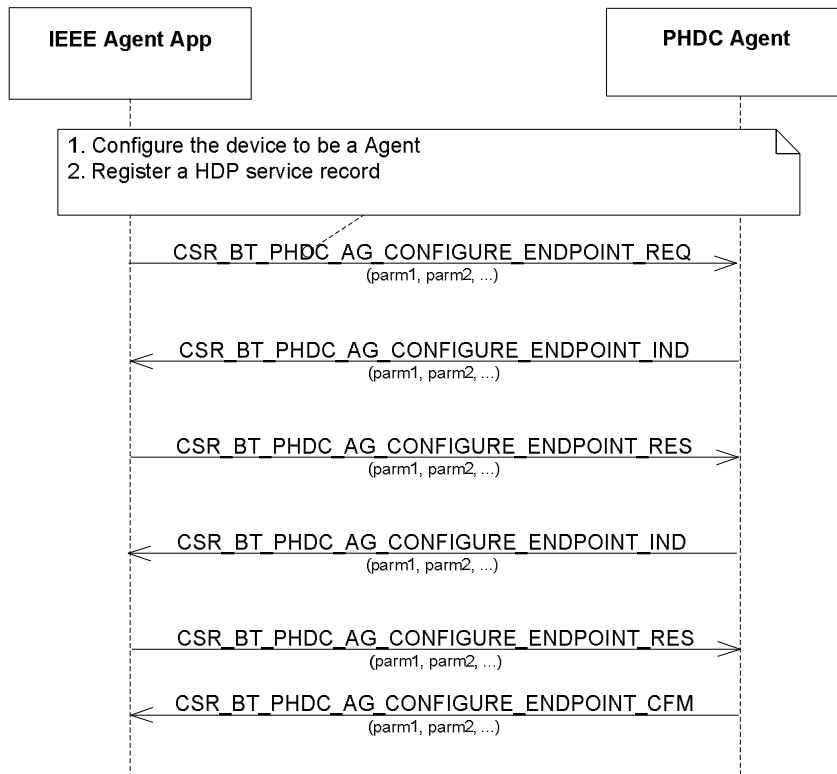


Figure 3: Configuration Sequence

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_CONFIGURE_ENDPOINT_IND
2	CsrBtMdepld	mdepld	<p>An identifier that identifies one or more logical functions advertised in the service record. It allows HDP to multiplex communication to various data layer end points.</p> <p>This is generated by HDP service layer.</p> <p>The response will indicate the functions the application wishes to advertise on this endpoint.</p>

Table 2: Parameters in CSR_BT_PHDC_AG_CONFIGURE_ENDPOINT_IND

#	Type	Argument	Description
1	CsrBtMdepId	mdepId	MdepId set to mdepId received in CsrBtPhdcAgConfigureEndPointInd
2	CsrBtMdepDataType	datatype	Identifies the Device Data Specialization advertised on this end point
3	CsrBtMdepRole	role	Indicates whether this MDEP is a Source of the identified data type, or a Sink
4	CsrUtf8String*	description	Description of the end-point
5	CsrBool	reuseMdepId	Indicates if the device data specialisation should be multiplexed on the mdepId.

Table 3: Arguments for CsrBtPhdcAgConfigureEndPointResSend

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_CONFIGURE_ENDPOINT_CFM
2	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
3	CsrBtResultSupplier	resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in csr_bt_result.h

Table 4: Parameters in a CSR_BT_PHDC_AG_CONFIGURE_ENDPOINT_CFM primitive

3.2 Device Configure

The API's described in this section are used for configuring the Agent with health device objects.

Sending a CSR_BT_PHDC_AG_CONFIGURE_MDS_REQ will configure the MDS object attributes of the health device. There is no confirmation for this message.

#	Type	Argument	Description						
1	CsrUInt8	supportedDeviceSpecializations	<div>Bitmap indicating the supported device specializations.</div> <table><tr><td>Bit8 to Bit 3</td><td>Bit2</td><td>Bit 1</td></tr><tr><td></td><td>BPM</td><td>WS</td></tr></table> <div>Bit 3 to Bit 8 are not used.</div> <div>Setting of particular bit indicates support of corresponding device specialization.</div>	Bit8 to Bit 3	Bit2	Bit 1		BPM	WS
Bit8 to Bit 3	Bit2	Bit 1							
	BPM	WS							
2	CsrCharString*	manufacturer	String containing Manufacturer name. This string should have even length.						
3	CsrCharString*	modelName	String containing model number. This string should have even length.						
4	CsrUInt8	systemId[8]	<div>64 bit System Id of the Agent device.</div> <div>- systemId [0], systemId [1] , systemId [2] – 24 Bits of MSB contains Organizationally Unique Identifier (OUI).</div> <div>- systemId [3] to systemId[7] - contains 40-bit manufacturer defined identifier.</div>						
5	CsrUInt16	devConfigId	Identification of Agent device configuration						
6	CsrCharString*	serialNumber	Serial Number of Agent device. This string should have even length.						
7	CsrCharString*	fwVersion	String containing component revisions, serial numbers etc in manufacture specific format. This string should have even length.						
8	CsrBtPhdcAgAbsoluteTime	absolutetime	Date and time of an Agent with resolution of 1/100th of a second						

Table 5: Arguments for CsrBtPhdcAgConfigureMdsReqSend

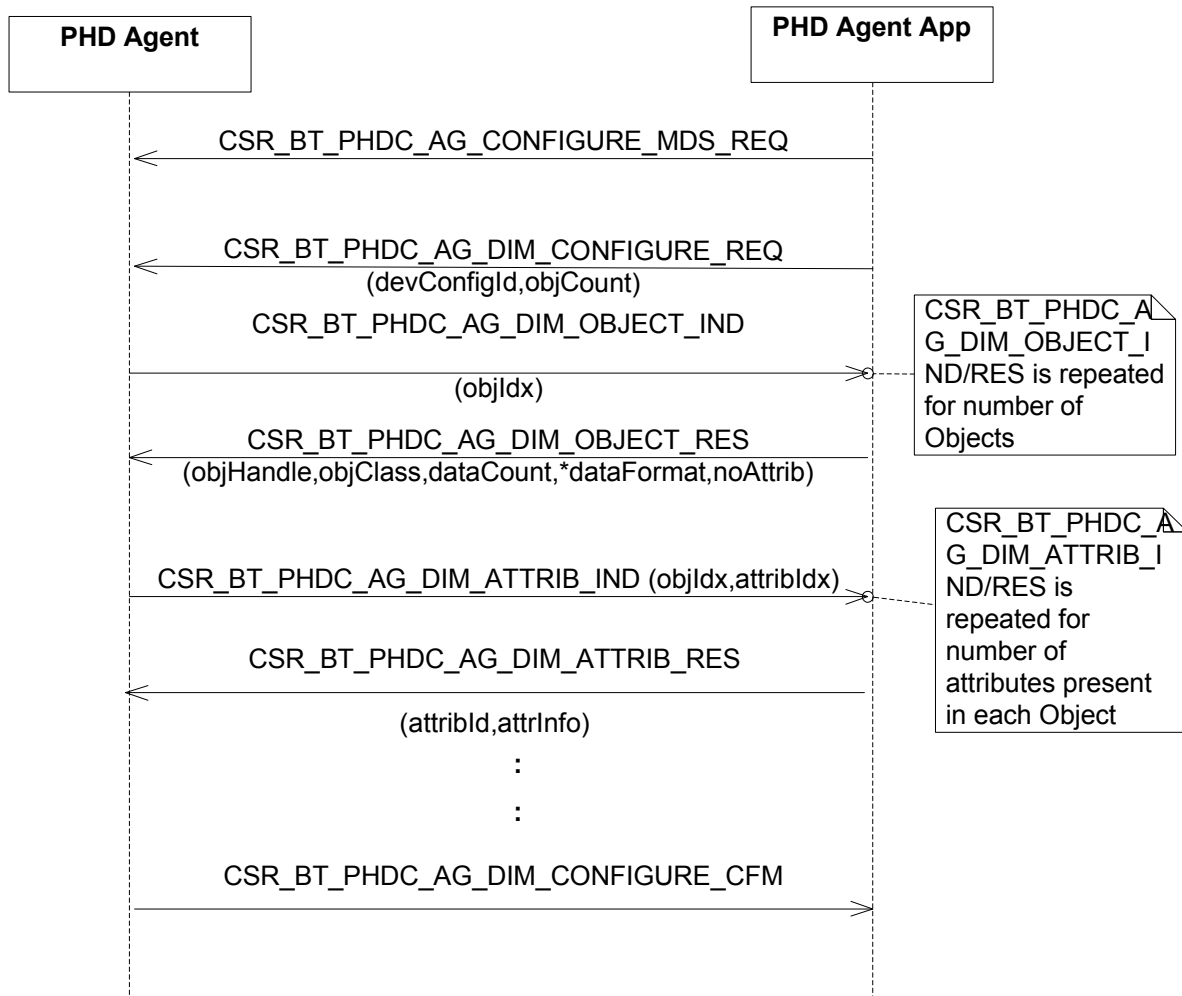


Figure 4: Device Configure Sequence

Sending a CSR_BT_PHDC_AG_DIM_CONFIGURE_REQ will configure the non-MDS object attributes of the health device. The usage of these API's would require knowledge about the domain information model(Refer IEEE 11073 – 20601 Sec 6).

#	Type	Argument	Description
1	CsrUInt16	devConfigId	Identification of Agent device configuration
2	CsrUInt16	objCount	Number of Objects

Table 6: Arguments for CsrBtPhdcAgDimConfigureReqSend

In case of standard configuration, CSR_BT_PHDC_AG_DIM_CONFIGURE_CFM is sent immediately as a response to CSR_BT_PHDC_AG_DIM_CONFIGURE_REQ.

In case of extended configuration for number of objects indicated in CSR_BT_PHDC_AG_DIM_CONFIGURE_REQ a phase of CSR_BT_PHDC_AG_DIM_OBJECT_IND/ CSR_BT_PHDC_AG_DIM_OBJECT_RES are used for enumerating object information and for number of attributes indicated in CSR_BT_PHDC_AG_DIM_OBJECT_RES a phase of CSR_BT_PHDC_AG_DIM_ATTRIB_IND/ CSR_BT_PHDC_AG_DIM_ATTRIB_RES are used for enumerating attribute information of each object.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_DIM_OBJECT_IND
2	CsrUInt16	objIndex	Index to object whose information is requested.

Table 7: Parameters in a CSR_BT_PHDC_AG_DIM_OBJECT_IND primitive

#	Type	Argument	Description
1	CsrUInt16	objIndex	Index of the Object
2	CsrUInt16	objClass	Class of the Object
3	CsrUInt16	objHandle	Handle of the Object
4	CsrUInt16	dataValueLength	Number of elements in dataValue array
5	CsrUInt16*	dataValue	Each element in dataValue represents a basic data type.
6	CsrUInt16	numAttrib	Number of Attributes present in this object

Table 8: Arguments for CsrBtPhdcAgDimObjectResSend

In Table above, fields "dataValueLength" and "dataValue" defines the format in which measurement data would be sent using CSR_BT_PHDC_AG_DATA_REQ API.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_DIM_ATTRIB_IND
2	CsrUInt16	objIndex	Index to object whose information is requested.
3	CsrUInt16	attribIndex	Index to attribute id within the object referred by objectIndex

Table 9: Parameters in a CSR_BT_PHDC_AG_DIM_ATTRIB_IND primitive

#	Type	Argument	Description
1	CsrUInt16	objIndex	Index of the Object
2	CsrUInt16	attribIndex	Index of the attribute ID
3	CsrUInt16	attribId	Attribute ID
4	CsrUInt16	attribInfoLength	Length of attribInfo pointer
5	CsrUInt8*	attribInfo	Attribute information as given for standard attributes defined in IEEE 11073-20601 specification 'attribInfo' is of type suitable for attribId.

Table 10: Arguments for CsrBtPhdcAgDimAttribResSend

Completion of configuration procedure for devices with extended device configuration is indicated by CSR_BT_PHDC_AG_DIM_CONFIGURE_CFM.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_DIM_CONFIGURE_CFM
2	CsrBtPhdcAgResult	resultCode	The result code of the operation.

Table 11: Parameters in a CSR_BT_PHDC_AG_DIM_CONFIGURE_CFM primitive

3.3 Device Capabilities

Sending CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_REQ will initiate a service search on the remote device for HDP-SINK matching the device specialisations set in the mask.

#	Type	Argument	Description
1	CsrBtDeviceAddr	deviceAddr	Bluetooth address of the remote device.
2	CsrUInt8	mdepDataTypeMask	Bit Mask of the Device Specializations.

Table 12: Arguments for CsrBtPhdcAgGetDeviceCapabReqSend

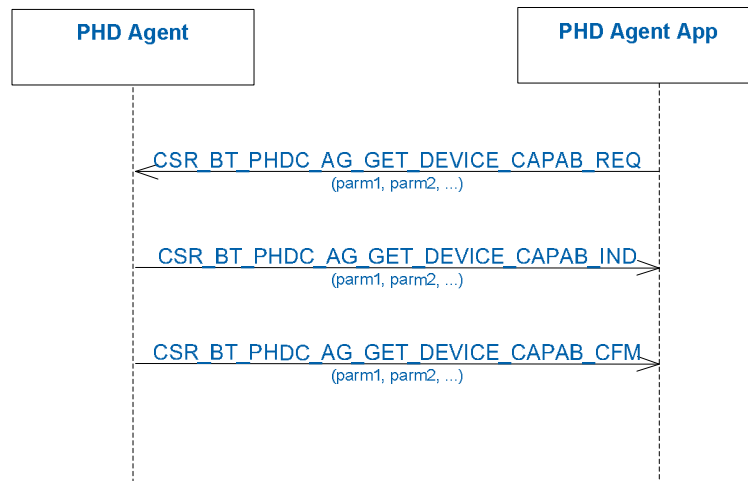


Figure 5: Device Search Sequence

For each service record that matches the search criteria a CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_IND is sent to the application

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_IND
2	CsrBtDeviceAddr	deviceAddr	Bluetooth address of the remote device.
3	CsrUInt32	psmIdentifier	The psmIdentifier is a unique Id generated for each connection. The higher 16 bits contain the Control PSM of the peer device and the lower 16 bits the Data PSM.
4	CsrCharString*	providerName	The ProviderName attribute is a string containing the name of the person or organization providing the service.
5	CsrCharString*	serviceName	The ServiceName is a string containing the name of the service represented by a service record.
6	CsrUInt8	supportedFeatureListLength	Number of elements in supportedFeatureList
6	CsrBtHdpMdep	supportedFeatureList	A list of MDEP endpoint ID and its associated features like MDEP Datatype, Role and Description.

Table 13: Parameters in a CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_IND primitive

The completion of the operation is notified to the application by CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_CFM. The parameters for the CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_CFM primitive are found in Table 14.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_CFM
2	CsrBtDeviceAddr	deviceAddr	Bluetooth address of the peer device.
3	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

#	Type	Argument	Description
4	CsrBtSupplier	resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in csr_bt_result.h

Table 14: Parameters in a CSR_BT_PHDC_AG_GET_DEVICE_CAPAB_CFM primitive

3.4 BT Link Establishment

Peer devices can initiate establishing an associate channel, this is indicated to Agent application by CSR_BT_PHDC_AG_ACCEPT_BT_LINK_IND. The device can accept or reject the BT connection by sending CSR_BT_PHDC_AG_ACCEPT_BT_LINK_RES. This procedure establishes an associate channel between devices.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_ACCEPT_BT_LINK_IND
2	CsrBtDeviceAddr	deviceAddr	Bluetooth address of the remote device.

Table 15: Parameters in CSR_BT_PHDC_AG_ACCEPT_BT_LINK_IND primitive

#	Type	Argument	Description
1	CsrBtDeviceAddr	deviceAddr	Bluetooth address of the remote device.
2	CsrBool	accept	TRUE – Accept the connection FALSE – Reject the connection
3	CsrUInt16	maxPacketLength	Maximum length of Application protocol data unit supported by the device for the connection.

Table 16: Arguments for CsrBtPhdcAgAcceptBtLinkResSend

3.5 Association

Sending the CSR_BT_PHDC_AG_ASSOCIATE_REQ will result in triggering the IEEE 11073 – 20601 association procedure and if required configuration procedure.

#	Type	Argument	Description
1	CsrUInt32	psmIdentifier	The Instance Id is a unique Id generated for each connection. The higher 16 bits contain the Control PSM of the peer device and the lower 16 bits the Data PSM.
2	CsrBtDeviceAddr	deviceAddr	Bluetooth address of the remote device.
3	CsrBtMdepId	mdepId	MdepId set to mdepId received in CsrBtPhdcConfigureEndPointInd
4	CsrUInt16	mdepDataType	This attribute is a 16-bit value, with the value taken from the Bluetooth Assigned Numbers [3] to identify the Device Data Specialization code.
5	CsrUInt16	maxPacketLength	Maximum length of Application protocol data unit supported by the device for the connection.
6	CsrUInt16	devConfigId	Identification of Agent device configuration

#	Type	Argument	Description
7	CsrtBool	release	<p>Agent device can support more than one configuration. During association procedure, manager device may indicate not to support the configuration included in association request for which agent could reply by sending the next configuration, this procedure continues until agent is exhausted with all available configurations. When the association fails with all configurations the association channel needs to be disconnected which is indicated by the flag "release"</p> <p>TRUE - Release associate channel FALSE – Do not Release associate channel.</p>

Table 17: Arguments for CsrBtPhdcAgAssociateReqSend

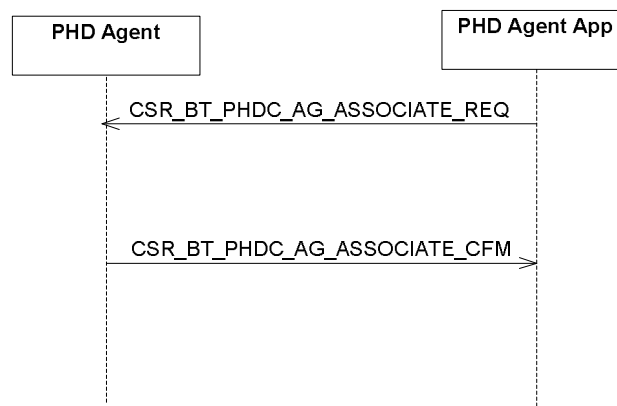


Figure 6 Association Sequence

Completion of associate procedure is indicated by CSR_BT_PHDC_AG_ASSOCIATE_CFM

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_ASSOCIATE_CFM
2	CsrUInt32	psmIdentifier	The psmIdentifier is a unique Id generated for each connection. The higher 16 bits contain the Control PSM of the peer device and the lower 16 bits the Data PSM.
3	CsrBtDeviceAddr	deviceAddr	Bluetooth address of the remote device.
4	CsrBtMdepld	mdepld	Mdepid set to mdepld received in CsrBtPhdcConfigureEndPointInd
5	CsrUInt16	mdepDataType	This attribute is a 16-bit value, with the value taken from the Bluetooth Assigned Numbers [3] to identify the Device Data Specialization code.
6	CsrUInt16	maxPacketLength	Maximum transmission unit supported by the device for the connection.
7	CsrUInt32	dataChannelId	Data Channel id set during association procedure
8	CsrBtPhdcAgResult	resultCode	The result code of the operation. Possible are defined in csr_bt_phdc_ag_prim.h

Table 18: Parameters in CSR_BT_PHDC_AG_ASSOCIATE_CFM primitive

3.6 Disassociation

Sending the CSR_BT_PHDC_AG_DISASSOCIATE_REQ will result in triggering IEEE 11073 – 20601 disassociation procedure. Disassociation is followed by releasing of BT link.

#	Type	Argument	Description
1	CsrUInt32	dataChannelId	Data Channel id set during association procedure
2	CsrUInt8	rlrqReason	Reason for disassociation <ul style="list-style-type: none"> - Normal - no-more-configurations - configuration-changed

Table 19: Arguments for CsrBtPhdcAgDisassociateReqSend function

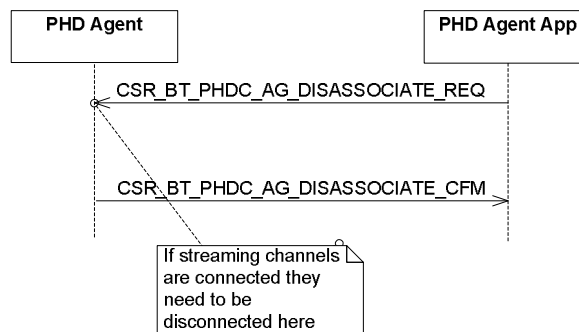


Figure 7: Disassociation Sequence

Completion of Disassociation procedure indicated to initiator by CSR_BT_PHDC_AG_DISASSOCIATE_CFM

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_DISASSOCIATE_CFM
	CsrUInt32	dataChannelId	Data Channel id set during association procedure
2	CsrBtDeviceAddr	deviceAddr	Bluetooth address of remote device.
3	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
4	CsrBtSupplier	resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in csr_bt_result.h

Table 20: Parameters in a CSR_BT_PHDC_AG_DISASSOCIATE_CFM primitive

In case of disassociation initiated by Manager, Agent is indicated by CSR_BT_PHDC_AG_DISASSOCIATE_IND.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_DISASSOCIATE_IND
2	CsrUInt32	dataChannelId	Data Channel id set during association procedure
3	CsrBtDeviceAddr	deviceAddr	Bluetooth address of remote device.

Table 21: Parameters in a CSR_BT_PHDC_AG_DISASSOCIATE_IND primitive

3.7 Suspend/Resume

Channel suspension procedure is used for saving power when application has sent all its measurements. This procedure would result in releasing the BT link without releasing the association.

When the application has more data to send it can use Resume procedure.

The application can suspend the Association channel using the `CsrBtPhdcAgSuspendReqSend` function. This suspend procedure will releases the association and disconnect all the Virtual and Association Channels established between the two devices. The arguments for the `CsrBtPhdcAgSuspendReqSend` function are found in Table 22.

#	Type	Argument	Description
1	CsrUInt32	dataChannelId	A 32 bit integer ID generated for each association whose lower 16 bits from the reliable MDL ID and the higher 16 bits the MCL ID of the corresponding connection.

Table 22 : Argument to CsrBtPhdcAgSuspendReqSend function

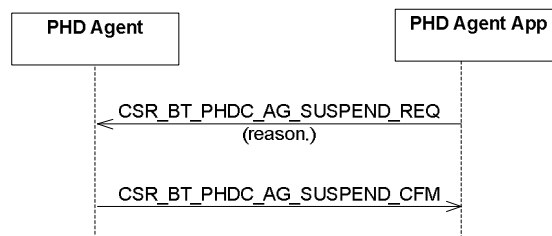


Figure 8: Suspend Sequence

Completion of suspension procedure is indicated by `CSR_BT_PHDC_AG_SUSPEND_CFM`.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to <code>CSR_BT_PHDC_AG_SUSPEND_CFM</code>
2	CsrUInt32	dataChannelId	A 32 bit integer ID generated for each association whose lower 16 bits from the reliable MDL ID and the higher 16 bits the MCL ID of the corresponding connection.
3	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == <code>CSR_BT_SUPPLIER_CM</code> then the possible result codes can be found in <code>csr_bt_cm_prim.h</code> . All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
4	CsrBtSupplier	resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in <code>csr_bt_result.h</code>

Table 23: Parameters in a CSR_BT_PHDC_AG_SUSPEND_CFM primitive

In case of suspension procedure initiated by Manager, Agent gets to `CSR_BT_PHDC_AG_SUSPEND_IND`

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to <code>CSR_BT_PHDC_AG_SUSPEND_IND</code>
2	CsrUInt32	dataChannelId	A 32 bit integer ID generated for each association whose lower 16 bits from the reliable MDL ID and the higher 16 bits the MCL ID of the corresponding connection.

Table 24: Parameters in a CSR_BT_PHDC_AG_SUSPEND_IND primitive

The application can resume the Association channel using the `CsrBtPhdcAgResumeReqSend` function.

#	Type	Argument	Description
1	CsrUInt32	dataChannelId	A 32 bit integer ID generated for each association whose lower 16 bits from the reliable MDL ID and the higher 16 bits the MCL ID of the corresponding connection.

Table 25: Argument to `CsrBtPhdcAgResumeReqSend` function

Completion of resume procedure is indicated by `CSR_BT_PHDC_AG_RESUME_CFM`

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to <code>CSR_BT_PHDC_AG_RESUME_CFM</code>
2	CsrUInt32	dataChannelId	A 32 bit integer ID generated for each association whose lower 16 bits from the reliable MDL ID and the higher 16 bits the MCL ID of the corresponding connection.
3	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == <code>CSR_BT_SUPPLIER_CM</code> then the possible result codes can be found in <code>csr_bt_cm_prim.h</code> . All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
4	CsrBtSupplier	resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in <code>csr_bt_result.h</code>

Table 26: Parameters in a `CSR_BT_PHDC_AG_RESUME_CFM` primitive

In case of resume procedure initiated by Manager, Agent gets to `CSR_BT_PHDC_AG_RESUME_IND`

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to <code>CSR_BT_PHDC_AG_RESUME_IND</code>
2	CsrUInt32	dataChannelId	A 32 bit integer ID generated for each association whose lower 16 bits from the reliable MDL ID and the higher 16 bits the MCL ID of the corresponding connection.
3	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == <code>CSR_BT_SUPPLIER_CM</code> then the possible result codes can be found in <code>csr_bt_cm_prim.h</code> . All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
4	CsrBtSupplier	resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in <code>csr_bt_result.h</code>

Table 27: Parameters in a `CSR_BT_PHDC_AG_RESUME_IND` primitive

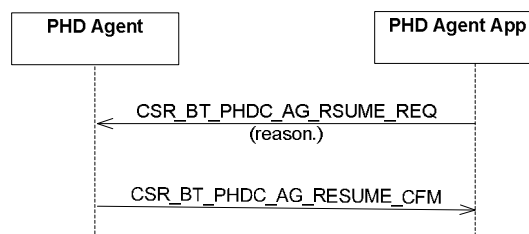


Figure 9: Resume Sequence

3.8 Sending Data

When an association channel is connected, it is possible to send data to the remote device. CSR_BT_PHDC_AG_DATA_REQ API is used for sending data. Data is sent by calling the CsrBtPhdcAgDataReqSend function.

Data transmission procedure by PHDC agent: In a measurement report, PHDC agent can include maximum of 25 measurements. Size of measurement report is also limited by the parameter maxPacketLength sent in CsrBtPhdcAgAssociateReqSend. At the reception of CSR_BT_PHDC_AG_DATA_REQ above checks are made and if any of condition is met the measurement report is sent to the remote device, once the confirmation is received same is indicated to application by CSR_BT_PHDC_AG_DATA_CFM.

In the sequence of sending measurements if application receives CSR_BT_PHDC_AG_DATA_CFM(SUCCESS) indicates that Agent has successfully transmitted the measurements and any pending measurements can be further sent by application.

#	Type	Argument	Description
1	CsrUInt32	dataChannelId	A 32 bit Id that identifies the channel on which data is sent
2	CsrUInt8	transmitFlag	Following values are used <ul style="list-style-type: none"> - START : First Measurement - CONTINUE : More measurements to follow - FINAL : Last measurement
3	CsrBtPhdcAgDataNode	data	Payload data

Table 28: Arguments for CsrBtPhdcAgDataReqSend function

#	Type	Argument	Description
1	CsrUInt16	objHandle	Handle of the Object
2	CsrUInt16	personId	Person Id <ul style="list-style-type: none"> - valid only for multi person measurement reporting. - For non multiperson measurement should be set to CSR_BT_PHDC_AG_PERSON_ID_NA
3	CsrBtPhdcAgAbsoluteTime	time	Measurement Time
4	CsrUInt16	dataLength	Number of elements in "data". Derived from dataValueLength and dataValue fields of CsrBtPhdcDimObjectResSend.
5	CsrUInt16*	data	Payload data

Table 29: Members of structure CsrBtPhdcAgDataNode

#	Type	Argument	Description
1	CsrUInt8	century	
2	CsrUInt8	year	
3	CsrUInt8	month	
4	CsrUInt8	day	

#	Type	Argument	Description
5	CsrUInt8	hour	
6	CsrUInt8	minute	
7	CsrUInt8	second	
8	CsrUInt8	secondFractions	

Table 30: Members of structure CsrBtPhdcAgAbsoluteTime

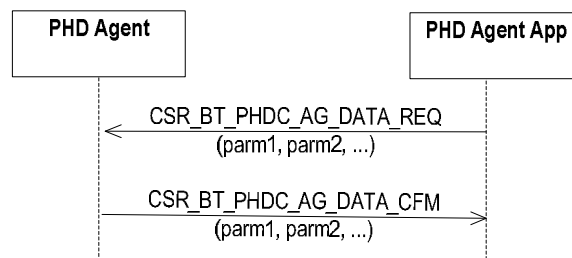


Figure 10: Data send Sequence

Completion of data request is indicated to the initiator by CSR_BT_PHDC_AG_DATA_CFM. Reception of CSR_BT_PHDC_AG_DATA_CFM also indicates the measurement has been received at the remote end.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_DATA_CFM
2	CsrUInt32	dataChannelId	A 32 bit Id that identifies the channel on which the data is received
3	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

Table 31: Parameters in a CSR_BT_PHDC_AG_DATA_CFM primitive

3.9 Change of Measurement Unit

Sending the CSR_BT_PHDC_AG_CHANGE_DIM_UNIT_CODE_REQ will update the measurement unit of the indicated attribute. This is used for modifying measurement unit without releasing the association. The updated attribute value is temporary and is valid only for the current association.

#	Type	Argument	Description
1	CsrUInt16	objHandle	Object Handle whose unit to be updated
2	CsrUInt16	unitCode	Changed unit value. For ex: Pounds, earlier measurement unit was in Kilograms

Table 32: Arguments of CsrBtPhdcAgChangeDimUnitCodeReqSend

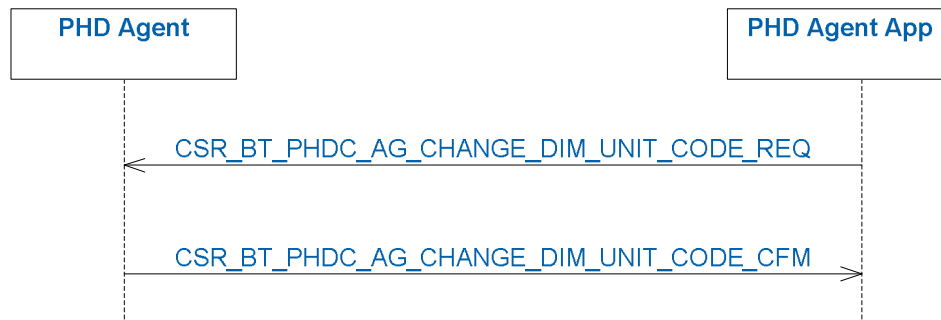


Figure 11: Measurement Unit update Sequence

Completion of Measurement unit procedure is indicated by CSR_BT_PHDC_AG_CHANGE_DIM_UNIT_CODE_CFM.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_CHANGE_DIM_UNIT_CODE_CFM
2	CsrUint16	objHandle	Object Handle whose unit is updated
3	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

Table 33: Parameters in a CSR_BT_PHDC_AG_CHANGE_DIM_UNIT_CODE_CFM primitive

3.10 Setup BT Link

Setup Bt link is indicated to application by CSR_BT_PHDC_AG_SETUP_BT_LINK_IND

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_SETUP_BT_LINK_IND
2	CsrBtDeviceAddr	deviceAddr	Bluetooth address of the remote device.
3	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
4	CsrBtSupplier	resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in csr_bt_result.h

Table 34: Parameters in CSR_BT_PHDC_AG_SETUP_BT_LINK_IND primitive

3.11 BT Link Release

Release of association channel during IEEE 11073 – 20601 disassociation procedure or while in IEEE 11073 – 20601 unassociated state is indicated to application by CSR_BT_PHDC_AG_BT_LINK_REL_IND.

#	Type	Argument	Description
1	CsrPhdcAgPrim	type	Signal identity – always set to CSR_BT_PHDC_AG_BT_LINK_REL_IND
2	CsrBtDeviceAddr	deviceAddr	Bluetooth address of the remote device.
3	CsrBtResultCode	resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
4	CsrBtSupplier	resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in csr_bt_result.h

Table 35: Parameters in CSR_BT_PHDC_AG_BT_LINK_REL_IND primitive

4 Document References

Document	Reference
Personal Health Device Communication – Application Profile, ISO/IEEE P11073 - 20601	[1]

Terms and Definitions

BlueCore®	Group term for CSR's range of Bluetooth wireless technology chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
PHDC	Personal Health Device Communication
MDS	Medical Device System

Document History

Revision	Date	History
1	26 SEP 11	Ready for release 18.2.0

TradeMarks, Patents and Licences

Unless otherwise stated, words and logos marked with [™] or [®] are trademarks registered or owned by CSR plc or its affiliates. Bluetooth[®] and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc. and licensed to CSR. Other products, services and names used in this document may have been trademarked by their respective owners.

The publication of this information does not imply that any licence is granted under any patent or other rights owned by CSR plc.

CSR reserves the right to make technical changes to its products as part of its development programme.

While every care has been taken to ensure the accuracy of the contents of this document, CSR cannot accept responsibility for any errors.

Life Support Policy and Use in Safety-critical Compliance

CSR's products are not authorised for use in life-support or safety-critical applications. Use in such applications is done at the sole discretion of the customer. CSR will not warrant the use of its devices in such applications.

Performance and Conformance

Refer to www.csrsupport.com for compliance and conformance to standards information.

5 Annex A: Usage of API's

This section illustrates the usage of some of API's described in this document.

Consider Blood pressure monitor health device with extended configuration having two objects with following details

Object type: Systolic/diastolic/MAP compound numeric object

Measurement information: Systolic, diastolic and MAP measurements to be sent along with the absolute time.

Data size:

- Systolic, diastolic and MAP occupies 2 bytes each
- absolute time occupies 8 bytes

Object type: Pulse rate numeric object

Measurement information: Pulse rate to be sent along with the absolute time

Data Size:

- Pulse rate occupies 4 bytes
- absolute time occupies 8 bytes

Figure below illustrates usage of API's with above objects information.

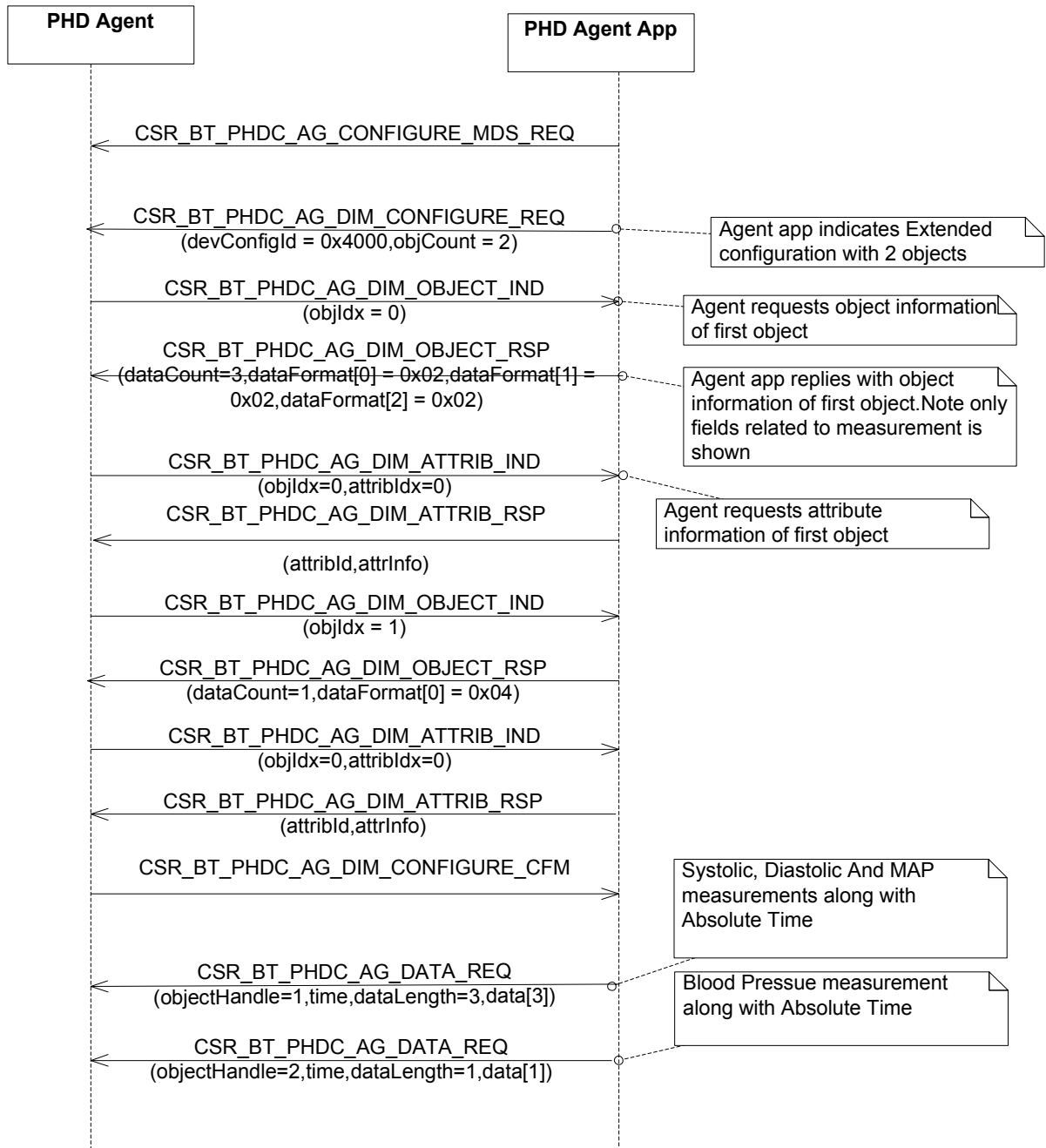


Figure 122 : Configuration and Data Transfer Sequence Example