BLUETOOTH® DOC	Date / Year-Month-Day 2012-04-03	Approved Adopted	Revision V10	Document No GLP_SPEC
Prepared By MED WG	E-mail Address med-main@bluetooth.org			N.B.

GLUCOSE PROFILE

Abstract:

This profile enables a device to connect and interact with a glucose sensor for use in consumer healthcare applications.

Revision History

Revision	Date (yyyy-mm-dd)	Comments
D09r00	2011-10-18	Initial Draft based upon state of Blood Pressure Profile from 18 October 2011.
D09r01	2011-10-20	Accepted all changes after significant updates. First version for MED WG review.
D09r02	2011-10-25	Accepted changes from Glucose Development call.
D09r03	2011-11-02	Revised according to reviews. Added procedure timeout handling section.
D09r04	2011-11-09	Accepted all changes. Readied for first BARB review. Updated based on initial BARB feedback.
D09r05	2011-11-14	Accepted all changes. Submitted for additional BARB review. Incorporated BARB feedback.
D09r06	2011-11-19	Accepted all changes. Re-submitted for BARB review. Incorporated feedback from BARB. Submitted for BARB vote.
D09r07	2011-11-23	Accepted all changes. Submitted for BARB vote. Fixed incorrect error code requirement. Added clarifications in 4.7.3 and 4.7.3.2 to avoid corner cases.
D09r08	2011-11-23	Accepted all changes. Submitted for BARB vote. Version approved by BARB for prototyping.
D10r00	2011-12-14	Updated to draft 1.0. Incorporated feedback from IOP, MED WG F2F and BARB F2F.
D10r01	2012-02-05	Accepted all changes. Submitted for BARB final review. Incorporated feedback from BARB review.
D10r02	2012-02-17	Accepted all changes. Incorporated further feedback from BARB.
D10r03	2012-02-23	Accepted all changes. BARB voting candidate.
D10r04	2012-02-23	Accepted all changes. BARB voting candidate.
V10	2012-04-03	Adopted by the Bluetooth SIG Board of Directors

Contributors

Name	Company
Robert Hughes	Intel Corporation
Badrinarayanan Krishnamoorthy	MindTree
Ray Strickland	Roche
Timothy Cook	Stonestreet One

Disclaimer and Copyright Notice

The copyright in this specification is owned by the Promoter Members of Bluetooth® Special Interest Group (SIG), Inc. ("Bluetooth SIG"). Use of these specifications and any related intellectual property (collectively, the "Specification"), is governed by the Promoters Membership Agreement among the Promoter Members and Bluetooth SIG (the "Promoters Agreement"), certain membership agreements between Bluetooth SIG and its Adopter and Associate Members (the "Membership Agreements") and the Bluetooth Specification Early Adopters Agreements (1.2 Early Adopters Agreements) among Early Adopter members of the unincorporated Bluetooth SIG and the Promoter Members (the "Early Adopters Agreement"). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to Bluetooth SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of Bluetooth SIG or a party to an Early Adopters Agreement (each such person or party, a "Member") is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by applicable law to Bluetooth SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the *Bluetooth* technology ("*Bluetooth* products") may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of *Bluetooth* products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Bluetooth Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Bluetooth products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. **NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.**

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserve the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.

Document Terminology

The Bluetooth SIG has adopted section 13.1 of the IEEE Standards Style Manual, which dictates use of the words ``shall'', ``should'', ``may'', and ``can'' in the development of documentation, as follows:

The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).

The use of the word *must* is deprecated and shall not be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

The use of the word *will* is deprecated and shall not be used when stating mandatory requirements; *will* is only used in statements of fact.

The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

The term *Reserved for Future Use (RFU)* is used to indicate Bluetooth SIG assigned values that are reserved by the Bluetooth SIG and are not otherwise available for use by implementations.

Table of Contents

1 Introduction

The Glucose Profile is used to enable a device to obtain glucose measurement and other data from a Glucose Sensor that exposes the Glucose Service [1].

1.1 Profile Dependencies

This profile requires the Generic Attribute Profile (GATT).

1.2 Conformance

If conformance to this profile is claimed, all capabilities indicated as mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth qualification program.

1.3 Bluetooth Specification Release Compatibility

This specification is compatible with any Bluetooth Core Specification [2] that includes the Generic Attribute Profile (GATT) and the Bluetooth Low Energy Controller portion of the core specification.

2 Configuration

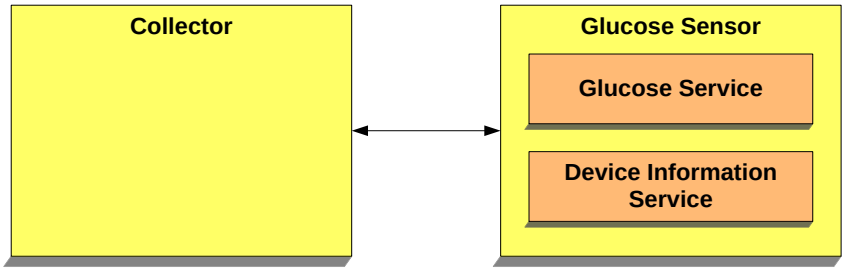
2.1 Roles

The profile defines two roles: Glucose Sensor and Collector. The Glucose Sensor is the device that measures the Glucose level concentration and the Collector is the device that receives the Glucose measurement and other related data from a Glucose Sensor.

- The Glucose Sensor shall be a GATT Server.
- The Collector shall be a GATT Client.

2.2 Role/Service Relationships

The following diagram shows the relationships between services and the two profile roles.



Note: Profile roles are represented by yellow boxes and services are represented by orange boxes.

A Glucose Sensor instantiates the Glucose Service [1] and the Device Information Service [3].

2.3 Concurrency Limitations and Restrictions

There are no concurrency limitations or restrictions for the Collector or Glucose Sensor roles imposed by this profile.

2.4 Topology Limitations and Restrictions

The Glucose Sensor shall use the GAP Peripheral role.

The Collector shall use the GAP Central role.

2.5 Transport Dependencies

This profile shall operate over an LE transport only. For BR/EDR (and HS) the Health Device Profile [4] is to be used.

3 Glucose Sensor Role Requirements

The Glucose Sensor shall instantiate one and only one Glucose Service [1].

The Glucose Service shall be instantiated as a «Primary Service».

The Glucose Sensor shall instantiate the Device Information Service [3].

Service	Glucose Sensor
Glucose Service	M
Device Information Service	M

Table 3.1: Glucose Sensor Service Requirements

See section 5.1 and section 6.1 for additional requirements for the Glucose Sensor role.

3.1 Incremental Glucose Service Requirements

This section describes additional Glucose Sensor requirements and recommendations beyond those defined in the Glucose Service.

3.1.1 Service UUIDs AD Type

While in a GAP Discoverable Mode for initial connection to a Collector, the Glucose Sensor should include the Glucose Service UUID defined in [5] in the Service UUIDs AD type field of the Advertising Data. This enhances the user experience as a Glucose Sensor may be identified by the Collector before initiating a connection.

3.1.2 Local Name AD Type

For enhanced user experience a Glucose Sensor should include the Local Name (containing either the complete or shortened value of the Device Name characteristic as defined in [2]) in its Advertising Data or Scan Response Data.

3.1.3 Writable GAP Device Name characteristic

The Glucose Sensor may support the write property for the Device Name characteristic in order to allow a Collector to write a device name to the Glucose Sensor.

3.1.4 Target Address AD Types

For enhanced user experience a Glucose Sensor that supports multiple bonds may include either the Public Target Address AD Type or Random Target Address AD Type (see section 5.1.5) in its Advertising Data or Scan Response Data. If a Glucose Sensor does not support multiple bonds, the Glucose Sensor shall not use a Target Address AD Type. As defined in the Glucose Service, the value of the Multiple Bond Supported bit of the Glucose Feature characteristic is to be set according to the Glucose Sensor's functionality so a Collector can determine if the Glucose Sensor does not support a Target Address AD Type (i.e. if the Multiple Bond Supported bit is 0) or potentially supports a Target Address AD Type (i.e. if the Multiple Bond Supported bit is 1). In the following sections the words "a Target Address AD Type" are intended to refer to either of the defined Target Address AD Types.

Sensor implementers should carefully consider the possible impact on the privacy of the user when the Sensor uses one of the Target Address AD Types. Although the use of a random address does make identification more difficult, if the address is used for long periods of time it could be associated with the user.

3.2 Incremental Device Information Service Requirements

The table below shows additional requirements and recommendations beyond those defined in the Device Information Service.

Device Information Service Characteristic	Requirement
Manufacturer Name String	M
Model Number String	M
System ID	M

Table 3.2: Device Information Service Requirements

Characteristics in this service may be transcoded by the Collector for use in an ISO/IEEE 11073 ecosystem. See the Personal Health Devices Transcoding White Paper [6] for more information. Since strings in this service are encoded as UTF-8, and ISO/IEEE 11073-20601a [7] specifies that strings are encoded as ASCII printable characters (a subset of UTF-8), characters used in string characteristics that are to be transcoded for use in an ISO/IEEE 11073 ecosystem must be restricted to the printable ASCII character set in order to ensure that the strings can be correctly displayed.

If the ISO/IEEE 11073-20601 specification is updated in the future to include UTF-8 support, implementers should consider the impact of using non-ASCII characters on backward compatibility. Note: The Personal Health Devices Transcoding White Paper [6] recommends that characters outside of the printable ASCII range are translated to characters inside of the printable ASCII range as appropriate.

4 Collector Role Requirements

The Collector shall support the Glucose Service [1].

The Collector may support the Device Information Service [3].

Service	Collector
Glucose Service	M
Device Information Service	O

Table 4.1: Collector Service Requirements

This section describes the profile procedure requirements for a Collector.

Profile Requirement	Section	Support in Collector
Service Discovery	4.2	M
- Glucose Service Discovery	4.2.1	M
- Device Information Service Discovery	4.2.2	O
Characteristic Discovery	4.3	M
- Glucose Service Characteristic Discovery	4.3.1	M
- Device Information Service Characteristic Discovery	4.3.2	O
Glucose Measurement	4.4	M
Glucose Measurement Context	4.5	M
Glucose Feature	4.6	M
Record Access Control Point	4.7	M

Table 4.2: Collector Requirements

4.1 GATT Sub-Procedure Requirements

Requirements in this section represent a minimum set of requirements for a Collector (Client). Other GATT sub-procedures may be used if supported by both Client and Server.

Table 4.3 summarizes *additional* GATT sub-procedure requirements beyond those required by all GATT Clients.

GATT Sub-Procedure	Collector (Client) Requirements
Discover All Primary Services	C.1
Discover Primary Services by Service UUID	C.1
Discover All Characteristics of a Service	C.2
Discover Characteristics by UUID	C.2
Discover All Characteristic Descriptors	M
Notifications	M
Read Characteristic Descriptors	M
Write Characteristic Descriptors	M

Table 4.3: Additional GATT Sub-Procedure Requirements

C.1: Mandatory to support at least one of these service discovery sub-procedures.

C.2: Mandatory to support at least one of these characteristic discovery sub-procedures.

4.2 Service Discovery

In order for the Collector to discover the characteristics of the Glucose Service, it shall perform primary service discovery using either the GATT *Discover All Primary Services* sub-procedure or the GATT *Discover Primary Services by Service UUID* sub-procedure. Recommended fast connection parameters and procedures for connection establishment are defined in section 5.2.4.

4.2.1 Glucose Service Discovery

The Collector shall discover the Glucose Service.

4.2.2 Device Information Service Discovery

The Collector may discover the Device Information Service.

4.3 Characteristic Discovery

As required by GATT, the Collector must be tolerant of additional optional characteristics in the service records of services used with this profile.

4.3.1 Glucose Service Characteristic Discovery

The Collector shall perform either the GATT *Discover All Characteristics of a Service* sub-procedure or the GATT *Discover Characteristics by UUID* sub-procedure in order to discover the characteristics of the service.

The Collector shall perform the GATT *Discover All Characteristic Descriptors* sub-procedure in order to discover the characteristic descriptors described in the following sections.

4.3.1.1 Glucose Measurement Characteristic

The Collector shall discover the Glucose Measurement characteristic.

The Collector shall discover the *Client Characteristic Configuration* descriptor of the Glucose Measurement characteristic.

4.3.1.2 Glucose Measurement Context Characteristic

The Collector shall discover the Glucose Measurement Context characteristic.

The Collector shall discover the *Client Characteristic Configuration* descriptor of the Glucose Measurement Context characteristic.

4.3.1.3 Glucose Feature Characteristic

The Collector shall discover the Glucose Feature characteristic.

4.3.1.4 Record Access Control Point Characteristic

The Collector shall discover the Record Access Control Point characteristic.

The Collector shall discover the *Client Characteristic Configuration* descriptor of the Record Access Control Point characteristic.

4.3.2 Device Information Service Characteristic Discovery

The Collector may discover the characteristics of the Device Information Service.

In order for the Collector to discover the characteristics of the Device Information Service, it shall use either the GATT *Discover All Characteristics of a Service* sub-procedure or the GATT *Discover Characteristics by UUID* sub-procedure to discover all characteristics of the service.

4.4 Glucose Measurement

Before performing any Record Access Control Point procedure, the Collector shall configure the Glucose Measurement characteristic for notifications (i.e. via the *Client Characteristic Configuration* descriptor).

When a Collector requires Glucose data it shall follow the Record Access Control Point procedures described in section 4.7.

The Collector shall determine the contents of the Glucose Measurement characteristic structure based on the contents of the Flags field. This allows the Collector to determine the unit of the Glucose Concentration field and whether or not optional fields defined by the characteristic are present.

The Collector shall be able to accept both units of the Glucose Concentration field (i.e. in base units of kg/L and mol/L).

If the Collector receives a Glucose Measurement characteristic with bits or values of the fields listed below that are designated as Reserved for Future Use (RFU) in [5], it shall continue to process the other fields of the Glucose Measurement characteristic normally.

- Flags field
- Type nibble of Type-Sample Location field
- Sample Location nibble of Type-Sample Location field
- Sensor Status Annunciation field

When the implementation receives a Glucose Measurement characteristic with additional, unrecognized, octets, the Collector behavior shall be identical to the Collector behavior when only recognized octets are received. This is to enable compatibility with future Glucose Service updates for the case where available octets in the characteristic are specified for optional use. What the Collector does with the additional, unrecognized, octets is left to the implementation.

4.5 Glucose Measurement Context

Before performing any Record Access Control Point procedure, the Collector shall (if supported by the Glucose Sensor) configure the Glucose Measurement Context characteristic for notifications (i.e., via the *Client Characteristic Configuration* descriptor).

When a Collector requires Glucose data it shall follow the Record Access Control Point procedures described in section 4.7.

The Collector shall determine the contents of the Glucose Measurement Context characteristic structure based on the contents of the Flags field. This allows the Collector to determine the unit of the Medication field and whether or not optional fields defined by the characteristic are present.

The Collector shall support both units of the Medication field (i.e. in base units of kilograms and liters).

If the Collector receives a Glucose Measurement Context characteristic with values of the fields listed below that are designated as Reserved for Future Use (RFU) in [5], it shall continue to process the other fields of the Glucose Measurement Context characteristic normally.

- Flags field
- Extended Flags field
- Carbohydrate ID field
- Meal Field
- Tester nibble of Tester-Health field
- Health nibble of Tester-Health field
- Medication ID field

When the implementation receives a Glucose Measurement Context characteristic with additional, unrecognized, octets, the Collector behavior shall be identical to the Collector behavior when only recognized octets are received. This is to enable compatibility with future Glucose Service updates for the case where available octets in the characteristic are specified for optional use. What the Collector does with the additional, unrecognized, octets is left to the implementation.

4.6 Glucose Feature

The Collector shall read the Glucose Feature characteristic to determine the supported features of the Glucose Sensor in order to interpret the bits in the Sensor Status Annunciation field of the Glucose Measurement characteristic. For example if the Low Battery Detection During Measurement feature is not supported, then the Device Battery Low at Time of Measurement bit in the Sensor Status Annunciation field has no meaning. On the other hand, if the Low Battery Detection During Measurement feature is supported, then the Device Battery Low at Time of Measurement bit will indicate that the sensor battery was detected to be low during a given measurement or not.

If the Low Battery Detection During Measurement Supported bit is set to 0 (False), the Collector shall ignore the Device Battery Low at Time of Measurement bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the Sensor Malfunction Detection Supported bit is set to 0 (False), the Collector shall ignore the Sensor Malfunction or Faulting at Time of Measurement bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the Sensor Sample Size Supported bit is set to 0 (False), the Collector shall ignore Sample Size for Blood or Control Solution Insufficient at Time of Measurement bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the Sensor Strip Insertion Error Detection Supported bit is set to 0 (False), the Collector shall ignore the Strip Insertion Error bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the Sensor Strip Type Error Detection Supported bit is set to 0 (False), the Collector shall ignore the Strip Type Incorrect For Device bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the Sensor Result High-Low Detection Supported bit is set to 0 (False), the Collector shall ignore the Sensor Result Higher Than the Device Can Process bit and the Sensor Result Lower Than the Device Can Process bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the Sensor Temperature High-Low Detection Supported bit is set to 0 (False), the Collector shall ignore the Sensor Temperature Too High for Valid Test/Result at Time of Measurement bit and the Sensor Temperature Too Low for Valid Test/Result at Time of Measurement bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the Sensor Read Interrupt Detection Supported bit is set to 0 (False), the Collector shall ignore the Sensor Read Interrupted Because Strip Was Pulled Too Soon at Time of Measurement bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the General Device Fault Supported bit is set to 0 (False), the Collector shall ignore the General Device Fault Has Occurred in the Sensor bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the Time Fault Supported bit is set to 0 (False), the Collector shall ignore the Time Fault Has Occurred in the Sensor and Time May Be Inaccurate bit of the Sensor Status Annunciation field of the Glucose Measurement characteristic.

If the Multiple Bond Supported bit is set to 0 (False), the Collector can determine that the Glucose Sensor supports only a single bond. Otherwise the Collector can determine that the Glucose supports multiple bonds.

Whether a bit is defined as static during the lifetime of the device (i.e. static permanently or until Service Changed is indicated) or static during a connection, is defined on a bit by bit basis in Table 3.2 of [1] (Static Requirements for Glucose Feature Bits).

If the Collector reads Glucose Feature characteristic bits that are set and yet are designated as Reserved for Future Use (RFU) in [5], it shall ignore those bits and continue to operate normally as if the bits were not set.

4.7 Record Access Control Point

Before performing any Record Access Control Point procedure, the Collector must configure the Record Access Control Point (RACP) characteristic for indications (i.e. via the *Client Characteristic Configuration* descriptor).

The Collector may perform a write to the Record Access Control Point to request a desired procedure. A procedure begins when the Collector writes the RACP to perform some desired action and ends when either a *Response Code* or *Number of Stored Records Response* RACP indication is received by the Collector.

If the procedure performed is the *Report Stored Records* procedure the procedure may contain one or more Glucose Measurement characteristic notifications and optionally one or more Glucose Measurement Context characteristic notifications between the write to the RACP characteristic that began the procedure and the *Response Code* RACP indication that ends the procedure.

If the Collector attempts to perform any defined Record Access Control Point procedure other than the *Abort Operation* procedure before a previous procedure is complete and receives a GATT Error Response with the error code set to *Procedure Already in Progress,* the Collector shall wait until the current RACP procedure completes before starting a new procedure.

If the Collector attempts to request any defined Record Access Control Point procedure before it has configured the Glucose Measurement characteristic for notifications, the Glucose Measurement Context characteristic for notifications (if the Glucose Measurement Context characteristic is supported by the Sensor) or the Record Access Control Point characteristic for indications (all via the appropriate *Client Characteristic Configuration* descriptor) as required in previous sections, then the Sensor will transmit a GATT Error Response with the error code set to *Client Characteristic Configuration Descriptor Improperly Configured*. This means that the Collector has not configured the Sensor correctly. The Collector must properly configure these characteristics for notifications or indications where appropriate, as defined by other portions of this document.

When a Collector requires a connection to a Glucose Sensor to receive Glucose data it shall follow the connection procedures described in section 5.2.

See section 9 for an example of how the RACP can be used in the context of the Glucose Profile.

4.7.1 Record Definition

Within the context of the Glucose Service, a record (also referred to as a patient record) consists of a Glucose Measurement characteristic and may or may not be followed by a corresponding Glucose Measurement Context characteristic. The Collector shall determine if a Glucose Measurement record includes a corresponding Glucose Measurement Context value based on the Context Information Flag in the Flags field. If the Context Information Flag for a Glucose Measurement record is set to 1, then a corresponding Glucose Measurement Context notification will follow the Glucose Measurement notification. See section 4.7.4.1 for recommendations on detecting and addressing single record errors.

4.7.2 RACP Procedure Requirements

The table below shows the requirements for the RACP procedures (Op Codes, Operators and Operands) in the context of this profile:

Op Code	Op Code Requirement	Operator	Operator Requirement	Operand		Operand Requirement
				Filter Type	Filter Parameters	
Report Stored Records	M	All records	M	No Operand Used		N/A
		Less than or equal to	O	Sequence Number	<maximum filter value>	C.1
				User Facing Time	<maximum filter value>	O
		Greater than or equal to	M	Sequence Number	<minimum filter value>	M
				User Facing Time	<minimum filter value>	O
		Within range of (inclusive)	O	Sequence Number	<minimum filter value>, <maximum filter value>	C.1
				User Facing Time	<minimum filter value>, <maximum filter value>	O
		First record	O	No Operand Used		N/A
		Last record	O	No Operand Used		N/A
Delete Stored Records	O	All records	C.2	No Operand Used		N/A
		Less than or equal to	O	Sequence Number	<maximum filter value>	C.1
				User Facing Time	<maximum filter value>	O
		Greater than or equal to	O	Sequence Number	<minimum filter value>	C.1
				User Facing Time	<minimum filter value>	O
		Within range of (inclusive)	O	Sequence Number	<minimum filter value>, <maximum filter value>	C.1
				User Facing Time	<minimum filter value>, <maximum filter value>	O
		First record	O	No Operand Used		N/A
		Last record	O	No Operand Used		N/A
Abort Operation	O	Null (0x00)	C.3	No Operand Used		N/A
Report	O	All records	M	No Operand Used		N/A

Number of Stored Records		Less than or equal to	O	Sequence Number	<maximum filter value>	C.1
				User Facing Time	<maximum filter value>	O
		Greater than or equal to	M	Sequence Number	<minimum filter value>	M
				User Facing Time	<minimum filter value>	O
		Within range of (inclusive)	O	Sequence Number	<minimum filter value>, <maximum filter value>	C.1
				User Facing Time	<minimum filter value>, <maximum filter value>	O
		First record	O	No Operand Used		N/A
		Last record	O	No Operand Used		N/A
Responses						
Op Code	**Op Code Require-ment**	**Operator**	**Operator Require-ment**	**Operand**		**Operand Require-ment**
Number of Stored Records Response	C.4	Null (0x00)	C.4	UINT16 containing number of records		C.4
Response Code	M	Null (0x00)	M	Request Op Code, Response Code Value		M

Table 4.4: RACP Procedure Requirements

C.1: If this Operator is supported, this Operand is mandatory for this Operator. See also Note 1.

C.2: If this Op Code is supported, this Operator is mandatory for this Op Code. See also Note 2.

C.3: Mandatory if the Abort Operation Op Code is supported, otherwise excluded.

C.4: Mandatory if the Report Number of Stored Records Op Code is supported, otherwise excluded.

Notes:

1. Support for a given Operand for one Op Code and Operator combination does not imply support of that Operand for other Op Code and Operator combinations.
2. Support for a given Operator for one Op Code does not imply support of that Operator for other Op Codes.
3. Where a filter type and filter parameters are used, the byte order for the Operand is specified in [1].

4.7.3 RACP Behavioral Description

The Collector shall write to the RACP characteristic using one of the supported Op Codes in Table 4.4 to request a Glucose Sensor to perform a procedure. This shall include an Operator and Operand that is valid within the context of that Op Code as defined in [1] .

The procedures including the use of Filter Types are described in the following sections.

The Collector shall be tolerant of the fact that the user may add or delete measurements *while* RACP procedures are taking place and also *between* two different RACP procedures. For example the Collector may perform the *Request Number of Stored Records* procedure with the Procedure Operator set to *All Stored Records*, to get the total number of records currently stored by the Sensor. However if the Collector then performs a *Report Stored Records* procedure with the Procedure Operator set to *All Stored Records,* it shall be tolerant of receiving a different number of records from this procedure than the Sensor reported having stored in the response to the *Report Number of Stored Records*. This is due to the fact that the Sensor may have taken new measurements and/or deleted stored measurements through user interaction, in the time between the *Request Number of Stored Records* procedure and the *Report Stored Records* procedure. This is used as an example, but the Collector shall be tolerant of this happening with any RACP procedure.

The Collector shall also be tolerant of the possibility that the Sensor, although required to maintain a Sequence Number continuum, may not be able to ensure a continuum in the case of any catastrophic hardware or software error that results in the Sequence Number being reset.

If the Sensor supports multiple bonds, a Collector shall be tolerant of the fact that other Collectors may alter the contents of the Sensor's measurement database. E.g. Collector #2 may delete records from the Sensor's database that Collector #1 will then never be able to retrieve. The handling of multiple bonds is vendor-specific. E.g. a Sensor may only allow certain Collectors (such as a doctor's computer) to use the Delete Store Records procedure.

4.7.3.1 Filter Types

Certain RACP procedures use a Filter Type that is used to determine the portion of the stored records on the Sensor that the procedure applies to. The format of the Filter Type and the RACP procedures that have a Filter Type operand is defined in [1]. The Collector may filter using either the *Sequence Number* filter or the *User Facing Time* filter.

The *Sequence Number* filter is used when a procedure should apply to specific Glucose Measurement patient record or a range of Glucose Measurement patient records. The Sequence Number that is used is obtained from the Sequence Number field in the Glucose Measurement characteristic.

The *User Facing Time* filter is used when a procedure should apply to a specific date and time or a range of dates and times.

The *User Facing Time* filter is provided to allow a mechanism for the Collector to perform RACP procedures with a given date or time (or a range of dates and times). However due to the fact that the time can be corrupted (due to hardware faults or other problems) the *Sequence Number* filter should be used in most cases. For example, the Collector can use the *Report Number of Stored Records* procedure to obtain all Glucose Measurement records that have been taken by a Sensor since the last record that the Collector has received. The Collector can do this by using the *Sequence Number* filter, with the Sequence Number set to the Sequence Number of the last Glucose

Measurement record that the Collector has received, incremented by one. Refer to section 3.1.1.2 of [1] for additional information.

4.7.3.2 Report Number of Stored Records procedure

To request the number of stored records from a Glucose Sensor, the Collector shall use the *Report Number of Stored Records* procedure. To receive the number based on filter criteria, an appropriate Operator and Operand shall be used. Refer to [1] for Operand requirements when used with a specific Operator and note that in some cases, no Operand is used.

The Collector shall wait for the *Number of Stored Records Response* RACP indication containing the number of stored records available in the Glucose Sensor as per the request. The *Number of Stored Records Response* RACP indication ends the *Report Number of Stored Records* procedure. In special circumstances, a Collector may require an abort of this procedure. See section 4.7.3.5 for details on that procedure.

If after requesting the number of stored records, the Collector receives a *Response Code* RACP indication with a *Response Code Value* that represents an error condition, see section 4.7.4 for general error handling procedures.

The value returned by the Number of Stored Records procedure is intended to be used either for the user interface on the Collector or to enable the Collector to acquire an estimate of the number of records it might receive to ensure it has sufficient resources. The Collector should not rely on this value as a means of determining if new records are available from the Sensor. The Collector should always request new records by filtering with a Sequence Number or User Facing Time incremented from a previous transfer session although Sequence Number is preferred as this is expected to be more robust.

4.7.3.3 Delete Stored Records procedure

To request deletion of stored records within the Glucose Sensor, the Collector shall use the *Delete Stored Records* procedure. To delete a selected set of stored records based on filter criteria, an appropriate Operator and Operand shall be used. Refer to [1] for Operand requirements when used with a specific Operator and note that in some cases, no Operand is used.

The Collector shall wait for the *Response Code* RACP Indication with the Response Code Value set to *Success* indicating successful deletion of records as per the request or for the procedure to time out according to the procedure time out operation described in section 4.7.5. In special circumstances, a Collector may require an abort of this procedure. See section 4.7.3.5 for details on the Abort Operation procedure.

If after requesting the deletion of stored records, the Collector receives a *Response Code* RACP indication with a *Response Code Value* that represents an error condition, see section 4.7.4 for general error handling procedures.

4.7.3.4 Report Stored Records procedure

To request the transfer of stored records from the Glucose Sensor, the Collector shall use the *Report Stored Records* procedure. To receive a selected set of stored records based on filter criteria, an appropriate Operator and Operand shall be used. Refer to [1] for Operand requirements when used with a specific Operator and note that in some

cases, no Operand is used. In special circumstances, a Collector may require an abort of this procedure. See section 4.7.3.5 for details on that procedure.

Once all patient records for a given request have been successfully indicated by the Glucose Sensor, the Glucose Sensor will send a *Response Code* RACP indication with the *Response Code Value* set to *Success*.

The Collector may also receive a *Response Code* RACP indication with the *Response Code Value* representing an error condition that occurred in processing the request. A description of specific error conditions and recommended procedures is described below. However, see section 4.7.4 for descriptions of general error conditions.

If after requesting stored records the Collector receives a *Response Code* RACP indication with the *Response Code Value* set to *No Records found*, this indicates that the Glucose Sensor does not have any stored records that meet the specified criteria.

If after requesting and receiving stored records the Collector receives a *Response Code* RACP indication with the *Response Code Value* set to *Record transfer incomplete*, this indicates that the Glucose Sensor was required to interrupt its data transfer before completion for an unspecified reason. If this occurs, the Collector should reattempt the data transfer using the same Op Code and Operator, but should modify the filter criteria in the Operand such that data already successfully received does not need to be retransmitted.

If a condition arises where a Collector is no longer able to receive the requested data, the Collector may request to abort the data transfer as described in section4.7.3.5.

4.7.3.5 Abort Operation procedure

To abort a procedure that a Collector initiated, the Collector shall use the *Abort Operation* Op Code with the Operator set to *Null* and no Operand.

The Collector shall then wait for the *Response Code* RACP indication with the *Response Code Value* set to *Success* indicating successful aborting of the procedure or for the procedure to time out according to the procedure time out operation described in section 4.7.5. Although Sensors are required to stop the data transfer after they have sent the Response Code Value of *Success*, they may still have some records queued for transfer. These will be sent and require acknowledgement from the Collector before the transfer will be fully terminated. The Collector may choose to process or ignore these additional records but shall be tolerant of this lag in the termination of the transfer.

The *Request Op Code* in the Operand of the *Response Code* RACP indication is used to determine if a *Response Code* RACP indication is received in response to an *Abort Operation* procedure, or the procedure that the *Abort Operation* is trying to abort. If the *Abort Operation* procedure is completed successfully then the Sensor shall send the *Response Code* RACP indication with the *Request Op Code* in the Operand set to *Abort Operation*, and shall not send any *Response Code* RACP indication for the aborted procedure.

The Collector may also receive a *Response Code* RACP indication with the *Request Op Code* in the Operand set to *Abort Operation* and the *Response Code Value* representing an error condition that occurred in processing the request. Though in practice not all *Response Code Values* may be returned for an *Abort Operation*

procedure, a Collector shall be able to handle receiving all defined *Response Code Values* in response this procedure. A description of specific error conditions and recommended procedures is described below. However, see section 4.7.4 for descriptions of general error conditions.

If after requesting the abort, the Collector receives a *Response Code* RACP indication with the *Request Op Code* in the Operand set to *Abort Operation* and the *Response Code Value* set to *Abort Unsuccessful*, this indicates that the Glucose Sensor is unable to process the abort.

4.7.4 General Error Handling

Other than error handling procedures that are specific to certain Op Codes, the following apply:

If the Collector writes an Op Code to the RACP characteristic that is unsupported by the Glucose Sensor, it will receive a *Response Code* RACP indication with the *Response Code Value* set to *Op Code not supported*.

If the Collector writes an Operator to the RACP characteristic that is invalid, it will receive a *Response Code* RACP indication with the *Response Code Value* set to *Invalid Operator*.

If the Collector writes an Operator to the RACP characteristic that is not supported by the Glucose Sensor, it will receive a *Response Code* RACP indication with the *Response Code Value* set to *Operator not supported*.

If the Collector writes an Operand to the RACP characteristic that is invalid, it will receive a *Response Code* RACP indication with the *Response Code Value* set to *Invalid Operand*.

If the Collector receives a *Response Code* RACP indication with the *Response Code Value* set to *Procedure not completed,* this indicates that the Sensor is unable to complete the procedure for some unknown reason, and the procedure shall be considered to have failed.

If the Collector writes a Filter Type within an Operand to the RACP characteristic that is not supported by the Glucose Sensor, it will receive a *Response Code* RACP indication with the *Response Code Value* set to *Operand Not Supported*.

4.7.4.1 Single Record Errors

In some circumstances, a received record may be invalid. This may be due to an implementation issue or a packet error caused for a variety of reasons. For increased robustness, it is recommended that Collectors check for invalid records. Some examples of checks are listed below:

- Check that if the Context Information Flag of Glucose Measurement characteristic is 0, no Glucose Measurement Context characteristic is present.
- Check that if the Context Information Flag of Glucose Measurement characteristic is 1, a Glucose Measurement Context characteristic is present.

- Check that if a Glucose Measurement characteristic is accompanied by a Glucose Measurement Context characteristic, the values in their respective Sequence Number fields are identical.
- Check that the value of the Flags field corresponds to the expected (minimum) length of the packet.

If an invalid record is detected, the Collector may re-request a single record using the RACP with the Operator set to 'within range of' and the Operand set to the Sequence Number Filter Type with the minimum value of the range and maximum value of the range both set to the Sequence Number value of the invalid record. If the record is retransmitted and still found to be invalid after a number of reattempts as left to the implementation, the Collector should discard the invalid record.

4.7.5 Procedure Timeout

In the context of the RACP characteristic, a procedure is started when the Sensor sends the response to the Collector's Write request for the RACP characteristic. The procedure is considered to be complete when the RACP characteristic is indicated with the Op Code set to *Response Code*.

A RACP procedure may consist of multiple notifications of the Glucose Measurement characteristic and optionally the Glucose Measurement Context characteristic. A RACP procedure may consist of multiple notifications of the Glucose Measurement characteristic and optionally the Glucose Measurement Context characteristic. The procedure is completed when the RACP characteristic is indicated. A procedure is considered to have timed out if a notification or an indication is not received within the ATT transaction timeout, defined as 30 seconds in Volume 2 Part F section 3.3.3 of [2], from the start of the procedure or from the last notification that was received as a result of this procedure.

If the link is lost while a RACP procedure is in progress then the procedure shall be considered to have timed out. See section 4.7.5.1 for handling this condition.

Thus a Collector shall start a timer, with the value set to the ATT transaction timeout, after the write response is received from the Sensor. The timer shall be restarted after every Glucose Measurement notification or Glucose Measurement Context notification is received. The timer shall be stopped when a RACP indication is received and the Op Code is set to *Response Code*. If the timer expires then the procedure shall be considered to have failed.

4.7.5.1 RACP Procedure Timeout Handling

If an RACP procedure times out (see section 4.7.5 for details of how this may occur) then no new RACP procedure shall be started by the Collector until a new link is established with the Sensor.

In the case of a procedure timeout during a *Report Stored Records* procedure, the Collector may consider the received Glucose Measurement records (and corresponding Glucose Measurement context values if they exist) to be valid. However the Collector shall not assume that the records received consist of all of the available records on the Sensor that match the filter criteria, i.e., the Collector shall assume that the records received during a *Report Stored Records* procedure that has timed out are an

incomplete set of records based on the filter criteria. The Collector may then perform the procedure again, when a new link is established, adjusting the filter criteria based on the records received during the aborted transfer.

4.8 Device Information Service Characteristics

The Collector may read the value of Device Information Service characteristics.

5 Connection Establishment

This section describes the connection establishment and connection termination procedures used by a Glucose Sensor and Collector in certain scenarios.

The following scenario description is informative:

Once configured by the Collector, a Glucose Sensor will typically remain powered off between uses and will only advertise and allow a Collector to connect when it is turned on by the user and has data to send. In the case where it is on and has data to send, , the Glucose Sensor will enter a GAP Connectable Mode and start advertising when it has data to send to the Collector. The Collector will typically execute a GAP connection establishment procedure such that it is scanning for the Glucose Sensor using a white list. When a connection is established, the Glucose Sensor sends one or more notifications and indications to the Collector. When the data transfer is complete the Glucose Sensor typically terminates the connection.

5.1 Glucose Sensor Connection Establishment

5.1.1 Connection Procedure for Unbonded Devices

This procedure is applicable when the Glucose Sensor connects to a Collector to which it is not yet bonded (i.e. initial connection). This is typically initiated through user interaction when a user intends to bond the Glucose Sensor with a Collector.

The Glucose Sensor should use the GAP *Limited Discoverable Mode* with connectable undirected advertising events when establishing an initial connection. The T_{GAP}(lim_adv_timeout) used during GAP *Limited Discoverable Mode* may be larger than the value specified in the section 16, Appendix A in the GAP specification [2], but the value shall be less than or equal to 180 seconds.

It is recommended to use the advertising interval parameters in Table 5.1 during the GAP *Limited Discoverable Mode*. The interval values in the first row are designed to attempt fast connection during the first 30 seconds; however, if a connection is not established within that time, the interval values in the second row are designed to reduce power consumption for devices that continue to advertise.

Advertising Duration	Parameter	Value
First 30 seconds (fast connection)	Advertising Interval	20 ms to 30 ms
After 30 seconds (reduced power)	Advertising Interval	1 s to 2.5 s

Table 5.1: Recommended Advertising Interval Values

Notwithstanding the above, the advertising interval and time to perform advertising should be configured with consideration for user expectations of connection establishment time.

The Glucose Sensor shall accept any valid values for connection interval and connection latency set by the Collector until service discovery and encryption is complete. Only after this has been completed should the Glucose Sensor request to change to the preferred connection parameters that best suits its use case.

If a connection is not established within a time limit defined by the Glucose Sensor, the Glucose Sensor may exit the GAP discoverable mode.

The Glucose Sensor shall be in a bondable mode during this procedure to optimize connecting to the Collector again using the procedure described in section 5.1.2.

If the Glucose Sensor has no data to transfer (or no further data to transfer), then after waiting for an idle connection timeout (see section 5.1.4), the Glucose Sensor should perform the GAP *Terminate Connection* procedure. This allows the Collector to perform any additional required actions (e.g. read Glucose Feature characteristic).

5.1.2 Connection Procedure for Bonded Devices

This procedure is applicable after the Glucose Sensor has bonded with the Collector using the connection procedure in section 5.1.1 and either when the user initiates a connection or autonomously when a new record has been taken.

A Glucose Sensor shall enter the GAP *Undirected Connectable Mode* either when commanded by the user to initiate a connection to a Collector or when a Glucose Sensor has one or more stored records to send to a previously connected Collector.

The Glucose Sensor should write the address of the target Collector in its White List and set its controller advertising filter policy to ‘process scan and connection requests only from devices in the White List’.

The Glucose Sensor should use the recommended advertising interval values shown in Table 5.1.

The advertising interval and time to perform advertising should be configured with consideration for user expectations of connection establishment time.

Once connected, the Glucose Sensor may request to change to the preferred connection parameters that best suits its use case.

If a connection is not established within a time limit defined by the Glucose Sensor, the Glucose Sensor may exit the GAP connectable mode.

If the Client Characteristic Configuration descriptor has been configured to enable notifications and the Collector does not perform a RACP procedure, then after waiting for an idle connection timeout (see section 5.1.4), the Glucose Sensor should perform the GAP *Terminate Connection* procedure. This allows the Collector to perform any additional required actions (e.g. read Glucose Feature characteristic).

Refer to section 5.1.5 for additional requirements related to support for multiple bonds.

5.1.3 Link Loss Reconnection Procedure

When a connection is terminated due to link loss, a Glucose Sensor should attempt to reconnect to the Collector by entering the GAP *Undirected Connectable Mode* using the recommended advertising interval values shown in Table 5.1.

5.1.4 Idle Connection

The Glucose Sensor should perform the GAP *Terminate Connection* procedure if the connection is idle for more than 5 seconds. For devices that support Man in the Middle

(MITM) protection, this duration may need to be longer to allow completion of the pairing sequence.

5.1.5 Multi-Bond Considerations

This section is applicable when a Glucose Sensor supports multiple bonds.

A Glucose Sensor supporting multiple bonds may advertise with the address of the target Collector(s) in a Target Address AD Type to enable bonded Collectors to determine if they are the intended recipient of the data. This feature is designed to avoid a situation where a bonded Collector unnecessarily responds to the Glucose Sensor advertisement intended for another bonded Collector. The applicable Public Target Address AD Type or Random Target Address AD Type format (see Table 5.2) should be used when including the address corresponding to Collector's address type. As stated in section 3.1.4, if the Glucose Sensor does not support multiple bonds a Target Address AD Type is not to be included in either the Advertising Data or the Scan Response Data.

If the Glucose Sensor uses a Target Address AD Type and was given a random address by that Collector, the Glucose Sensor shall use that address in the Random Target Address AD Type. A Glucose Sensor may include multiple target addresses in its Advertising Data or Scan Response Data although the maximum number possible is four due to data size restrictions. As described in section 5.1.2, the Glucose Sensor should also use a White List to avoid connection with an unwanted Collector.

Sensor implementations may, via the User Interface, enable the user to identify primary Collector(s) (i.e. personal Collector or Collectors used daily) and secondary Collector(s) (i.e. caregiver Collector or Collectors used monthly). For reduced power consumption, implementations should avoid advertising to secondary Collectors (i.e. infrequently available Collectors) unless triggered through user interaction.

Description	Information
Public Target Address	Multiples of 6 octets – 6 octets are required for each Device Address of the targeted Collector. This AD type is used when the target address is a public address. This data type shall exist only once.
Random Target Address	Multiples of 6 octets – 6 octets are required for each Device Address of the targeted Collector. This AD type is used when the target address is a random address. This data type shall exist only once.

Table 5.2: Target Address AD Types

5.2 Collector Connection Establishment

5.2.1 Connection Procedure for Unbonded Devices

This procedure is applicable for connection establishment when the Collector connects to a Glucose Sensor to which it is not yet bonded (i.e. initial connection). This is typically

initiated through user interaction when a user intends to bond the Collector with the Glucose Sensor.

A Collector should use the GAP *Limited Discovery* procedure to discover a Glucose Sensor.

The Collector should use the recommended scan interval and scan window values shown in Table 5.3. For the first 30 seconds (or optionally continuously for mains powered devices), the Collector should use the first scan window / scan interval pair to attempt fast connection. However, if a connection is not established within that time, the Collector should switch to one of the other scan window / scan interval options as defined below to reduce power consumption.

Once the Glucose Sensor is discovered, the Collector should initiate connection using the *Direct Connection Establishment* procedure.

Scan Duration	Parameter	Value
First 30 seconds (fast connection)	Scan Interval	30ms to 60ms*
	Scan Window	30ms
After 30 seconds (reduced power) - Option 1	Scan Interval	1.28s
	Scan Window	11.25ms
After 30 seconds (reduced power) - Option 2	Scan Interval	2.56s
	Scan Window	11.25ms

Table 5.3: Recommended Scan Interval and Scan Window Values

* A scan interval of 60ms is recommended when the Collector is supporting other operations to provide a 50% scan duty cycle versus 100% scan duty cycle.

Option 1 in the table above uses the same background scanning interval used in BR/EDR so the power consumption for LE will be similar to the power consumption used for background scanning on BR/EDR. Option 2 uses a larger background scanning interval (e.g. twice as long) than used in BR/EDR so the power consumption for LE will be less than the power consumption used for background scanning on BR/EDR. Connection times during background scanning will be longer with Option 2.

After the connection is established, the Collector shall bond with the Glucose Sensor during this procedure to optimize connecting to the Glucose Sensor again using the procedure in section 5.2.2. Once a bond is created, the Collector should write the address of the Glucose Sensor in the Collector controller's White List.

The Collector shall configure the Client Characteristic Configuration descriptor to enable notifications or indications as needed.

The Glucose Sensor typically terminates the connection if the Collector does not perform a RACP procedure after waiting for an idle connection timeout.

5.2.2 Connection Procedure for Bonded Devices

This procedure is applicable after the Collector has bonded with the Glucose Sensor using the connection procedure in section 5.2.1 and either when the user initiates a connection or autonomously when a Collector requires measurements from a Glucose Sensor.

A Collector may use one of the following GAP connection procedures based on its connectivity requirements:

- *General Connection Establishment Procedure*. The Collector may use this procedure when it requires measurements from one or more Glucose Sensors. This procedure allows a Collector to connect to a Glucose Sensor discovered during a scan without using the White List. The Collector should decode the Target Address AD Type to attempt to get the target address as described below in this section.
- *Selective Connection Establishment Procedure*. The Collector may use this procedure when it requires measurements from one or more Glucose Sensors. This procedure allows a Collector to connect to a Glucose Sensor discovered during a scan while using the White List. The Collector should decode the Target Address AD Type to attempt to get the target address as described below in this section.
- *Direct Connection Establishment Procedure*. The Collector may use this procedure when it requires measurements from a single (or specific) Glucose Sensor. The Collector may also use this procedure for link loss reconnection described in section 5.2.3.
- *Auto Connection Establishment Procedure*. The Collector may use this procedure when it requires measurements from one or more Glucose Sensors. This procedure will automatically initiate connection to a Glucose Sensor in the White List. The Collector should not use this procedure if one of the Glucose Sensors supports multiple bonds.

If the Collector receives an undirected connectable advertisement from a bonded Glucose Sensor while performing the General or Selective Connection Establishment Procedure, and the Glucose Sensor supports multiple bonds (i.e. the Multiple Bond Supported bit of the Glucose Feature characteristic is set to 1 meaning ‘Multiple Bonds supported’) see section 5.2.5 for multi-bond considerations and the use of target addresses. Otherwise a Target Address AD Type will not be present in either the Advertising Data or Scan Response Data.

The Collector should use the recommended scan interval and scan window values shown in Table 5.3. When initiating connection, the Collector should use the first scan window / scan interval pair to attempt fast connection. However, if a connection is not established within 30 seconds, the Collector should switch to one of the other scan window / scan interval options as defined in Table 5.3 to reduce power consumption.

If the Collector is in background scanning, it may use the scan window / scan interval Option 1 or Option 2 to reduce power consumption.

Notwithstanding the above, the Collector should use a scan window and scan interval suitable to its power and connection time requirements. Increasing the scan window increases the power consumption, but decreases the connection time.

The scan interval and scan window should be configured with consideration for user expectations of connection establishment time.

The Collector shall start encryption after each connection creation to verify the status of the bond. If encryption fails upon connection establishment (i.e. the bond no longer exists), the Collector must, after user interaction, re-bond, perform service discovery

(unless the Collector had previously determined that the Glucose Sensor did not have the <<Service Changed>> characteristic), and set the Glucose Sensor Client Characteristic Configuration descriptors again before using any of the services referenced by this profile in case the configuration was altered or lost.

The Glucose Sensor typically terminates the connection after it has no additional data to transfer.

5.2.3 Link Loss Reconnection Procedure

When a connection is terminated due to link loss, a Collector should attempt to reconnect to the Glucose Sensor using any of the GAP connection procedures with the parameters in Table 5.3.

5.2.4 Fast Connection Interval

To avoid very long service discovery and encryption times, the Collector should use the connection interval limits defined in Table 5.4 in the connection request.

Parameter	Value
Minimum Connection Interval	50 ms
Maximum Connection Interval	70 ms

Table 5.4: Recommended Fast Connection Interval Values

At any time a lower latency is required, for example to perform key refresh or encryption setup, this should be preceded with a connection parameter update to the minimum and maximum connection interval values defined in Table 5.4 and a connection latency of zero. This fast connection interval should be maintained as long as low latency is required. After that, it should switch to the preferred connection parameters as decided by the Glucose Sensor using the GAP *Connection Parameter Update* procedure.

5.2.5 Multi-Bond Considerations

If the Collector receives an undirected connectable advertisement from a bonded Glucose Sensor while performing the General or Selective Connection Establishment Procedure and the Glucose Sensor supports multiple bonds (i.e. the Multiple Bond Supported bit of the Glucose Feature characteristic is set to 1 meaning ‘Multiple Bonds supported’) then the Collector should decode the Target Address AD Type to attempt to get the target address. If supported by the Glucose Sensor, this will allow the Collector to determine the intended recipient of the advertisement.

If the Advertising Data or Scan Response Data contains a Target Address AD Type:

1. If the target address is the same as the Collector’s address, the Collector should initiate a connection to the Glucose Sensor using *Direct Connection Establishment* Procedure.
2. If the target address is not the same as the Collector’s address, the Collector should not initiate a connection to the Glucose Sensor.

If the Advertising Data and Scan Response Data do not contain a Target Address AD Type, the Collector may initiate Connection to the Sensor using *Direct Connection Establishment Procedure*.

6 Security Considerations

This section describes the security considerations for a Glucose Sensor and Collector.

6.1 Glucose Sensor Security Considerations

All supported characteristics specified by the Glucose Service shall be set to Security Mode 1 and either Security Level 2 or 3.

The Glucose Sensor shall bond with the Collector.

The Glucose Sensor shall use the SM *Slave Security Request* procedure to inform the Collector of its security requirements.

All characteristics specified by the Device Information Service that are relevant to this profile should be set to the same security mode and level as the characteristics in the Glucose Service.

6.2 Collector Security Considerations

The Collector shall bond with the Glucose Sensor.

The Collector shall accept any request by the Glucose Sensor for LE Security Mode 1 and either Security Level 2 or 3.

7 Acronyms and Abbreviations

Acronyms and Abbreviations	Meaning
AD	Advertising Data
BR/EDR	Basic Rate / Enhanced Data Rate
GAP	Generic Access Profile
GATT	Generic Attribute Profile
HS	High Speed
LE	Low Energy
MITM	Man in the Middle
RACP	Record Access Control Point
RFU	Reserved for Future Use
SM	Security Manager
UUID	Universally Unique Identifier

Table 7.1: Acronyms and Abbreviations

8 References

[1] Glucose Service
[2] Bluetooth Core Specification v4.0 or later
[3] Device Information Service v1.1 or later
[4] Health Device Profile v1.0 or later.
[5] Characteristic and Descriptor descriptions are accessible via the Bluetooth SIG Assigned Numbers
[6] Personal Health Devices Transcoding White Paper v1.2 or later
[7] ISO/IEEE Std 11073-20601™- 2008 Health Informatics - Personal Health Device Communication - Application Profile - Optimized Exchange Protocol - version 1.0 or later. This also includes ISO/IEEE Std 11073-20601a™- 2010 – Amendment 1.

9 Appendix A – Example of Record Access Control Point Usage

Below is an informative example showing the use of the RACP in the context of the Glucose Profile:

1. At 04 October 2011 12:40:00 pm (user-facing time internal to the Server i.e., Base Time + Time Offset), the Client requests records for the first time and requests the number of all records stored in the device.
    a. The Client writes Op Code 0x04 to request number of records with an Operator of 0x01 meaning “all records” and no Operand.
    b. The Server indicates back Op Code 0x05, an Operator of 0x00 (meaning “Null”) and Operand containing the number of all records (0x00F7 in this example)
2. Immediately after that, the Client requests a report of stored records.
    a. The Client writes Op Code 0x01 to request all records with an Operator of 0x01 meaning “all records” and no Operand.
    b. The Server notifies all records (Series of Glucose Measurement characteristics followed sometimes by Glucose Measurement Context characteristics) where the total number of Glucose Measurement characteristics totals 0x00F7.
    c. The Server indicates Op Code 0x06 with an Operator of 0x00 (meaning “Null”) and Operand of 0x01, 0x01 meaning “successful response to Op Code 0x01”.
    d. The Client stores the Sequence Number of the last received record for future use (0x00F7 since this was the first use and with the assumption in this example that the sequence number of the first record is 0x0001).
    e. Since this is a critical application, the Client performs some post-processing checks to make sure no major inconsistencies to the Base Time or Time Offset occurred. The Client also checks to see if any numbers in the sequence are missing.
3. Several days later, the Client requests a report of records since the last update.
    a. The Client writes Op Code 0x01 to request records with an Operator of 0x03 meaning “greater than or equal to” and an Operand set to Filter Type 0x01, 0x00F8) that is one number in the sequence more than the Sequence Number from the last record it received.
    b. The Server notifies all records that have accrued since the last request.
    c. The Server indicates Op Code 0x06 with an Operator of 0x00 (meaning “Null”) and Operand of 0x05, 0x01 meaning “successful response to Op Code 0x05”.