Voice Assistant Service

Bluetooth® Service Specification

- **Version:** v1.0
- **Version Date:** 2025-12-15
- **Prepared By:** Audio, Telephony, and Automotive Working Group

Abstract

The Voice Assistant Service (VAS) adds Voice Assistant (VA) functionality to Bluetooth® Low Energy (LE). This specification enables a client device to control and configure VA functionality.

Version History

Version Number	Date (yyyy-mm-dd)	Comments
v1.0	2025-12-15	Adopted by the Bluetooth SIG Board of Directors.

Acknowledgments

Name	Company
Hai Shalom	Apple Inc.
Hai Shalom	Google LLC
Scott Walsh	Plantronics Inc.
Andrew Estrada	Sony Group Corporation
Masahiko Seki	Sony Group Corporation
Jeff Solum	Starkey Hearing Technologies
Chris White	Dolby Laboratories, Inc.
Bjarne Klemmensen	Demant A/S
Chris Church	Qualcomm Technologies Inc.
Frank Yerrace	Microsoft Corporation
Georg Dickman	Sonova AG
Ahmad Rahmati	Google LLC
Rasmus Abildgren	Bose Corporation

Contents

1 Introduction 6
1.1 Language 6
1.1.1 Language conventions 6
1.1.1.1 Implementation alternatives 7
1.1.1.2 Discrepancies 7
1.1.2 Reserved for Future Use 7
1.1.3 Prohibited 7
1.2 Requirements in tables 7
1.2.1 Table requirements for unused cells 8
1.3 Conformance 8
1.4 Byte transmission order 8
2 Service 9
2.1 Service dependencies 9
2.2 Core Configuration dependencies 9
2.2.1 Core feature inter-layer dependencies 9
2.3 Dependencies on other specifications external to the Core 9
2.4 Attribute Protocol Application error codes 9
2.5 GATT inter-layer dependencies 9
2.6 Declaration 10
2.7 Behavior 10
3 Service characteristics 11
3.1 Voice Assistant Name characteristic 12
3.1.1 Voice Assistant Name format 12
3.1.2 Voice Assistant Name behavior 12
3.2 Voice Assistant UUID characteristic 12
3.2.1 Voice Assistant UUID format 12
3.3 VAS Control Point characteristic 13
3.3.1 VAS Control Point format 13
3.3.2 VAS Control Point behavior 14
3.3.2.1 Initialize VA Session 14
3.3.2.2 Start VA Session 14
3.3.2.3 Stop VA Session 14
3.4 Installed Location characteristic 15
3.4.1 Installed Location format 15
3.4.2 Installed Location behavior 15
3.5 Content Control ID characteristic 15
3.5.1 Content Control ID format 15
3.5.2 Content Control ID behavior 15
3.6 Voice Assistant Session State characteristic 15
3.6.1 Voice Assistant Session State format 15
3.6.2 Voice Assistant Session State behavior 16
3.6.2.1 Session Unavailable 17
3.6.2.2 Session Reset 17
3.6.2.3 Session Ready 17

3.6.2.4 Session Active 18
3.7 Voice Assistant Session Flag characteristic 18
3.7.1 Voice Assistant Session Flag format 18
3.7.2 Voice Assistant Session Flag behavior 18
3.7.2.1 Listening Now 19
3.7.2.2 Processing Now 19
3.7.2.3 Playback Now 19
3.8 Voice Assistant Supported Languages characteristic 19
3.8.1 Voice Assistant Supported Languages format 19
3.8.2 Voice Assistant Supported Languages behavior 19
3.9 Voice Assistant Supported Features characteristic 19
3.9.1 Voice Assistant Supported Features format 19
3.9.2 Voice Assistant Supported Features behavior 20
3.9.2.1 Session Flags Enabled 20

4 Acronyms and abbreviations 21

5 References 22

1 Introduction

A Voice Assistant (VA) is an intelligent tool that responds to spoken commands and helps users perform tasks or get information. This specification defines the Voice Assistant Service (VAS) that enables a Generic Attribute Profile (GATT) Client device (e.g., earbuds, a car kit, or a non-audio Bluetooth device) to control and interact with one or more VAs that reside on a GATT Server device (e.g., a smartphone or a laptop). This specification also defines the Generic Voice Assistant Service (GVAS) to provide easy access to a default VA.

1.1 Language

1.1.1 Language conventions

In the development of a specification, the Bluetooth SIG has established the following conventions for use of the terms "*shall*", "*mandatory*", "*shall not*", "*should*", "*should not*", "*may*", "*optional*", "*must*", and "*can*". In this Bluetooth specification, the terms in Table 1.1 have the specific meanings given in that table, irrespective of other meanings that exist.

Term	Definition
shall *or* mandatory	—used to express what is required by the specification and is to be implemented exactly as written without deviation
shall not	—used to express what is forbidden by the specification
should *or* may *or* optional	—not mandatory. Used to express either: 1. what is recommended by the specification without forbidding anything ("should") 2. what is permissible within the limits of the specification ("may" or "optional")
should not	—used to indicate that something is discouraged but not forbidden by the specification
must	—used to indicate either: 1. an indisputable statement of fact that is always true regardless of the circumstances 2. an implication or natural consequence if a separately-stated requirement is followed
can	—used to express a statement of possibility or capability

Table 1.1: Language conventions terms and definitions

Where more than one item is permitted but not required, the choices to include or support those items are independent from one another unless the specification explicitly states otherwise. Each item that is implemented shall be implemented exactly as written without deviation.

1.1.1.1 Implementation alternatives

When specification content indicates that there are multiple alternatives to satisfy specification requirements, if one alternative is explained or illustrated in an example it is not intended to limit other alternatives that the specification requirements permit.

1.1.1.2 Discrepancies

It is the goal of Bluetooth SIG that specifications are clear, unambiguous, and do not contain discrepancies. However, members can report any perceived discrepancy by filing an erratum and can request a test case waiver as appropriate.

1.1.2 Reserved for Future Use

Where a field in a packet, Protocol Data Unit (PDU), or other data structure is described as "Reserved for Future Use" (irrespective of whether in uppercase or lowercase), the device creating the structure shall set its value to zero unless otherwise specified. Any device receiving or interpreting the structure shall ignore that field; in particular, it shall not reject the structure because of the value of the field.

Where a field, parameter, or other variable object can take a range of values, and some values are described as "Reserved for Future Use," a device sending the object shall not set the object to those values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous; however, this does not apply in a context where the object is described as being ignored or it is specified to ignore unrecognized values.

When a field value is a bit field, unassigned bits can be marked as Reserved for Future Use and shall be set to 0. Implementations that receive a message that contains a Reserved for Future Use bit that is set to 1 shall process the message as if that bit was set to 0, except where specified otherwise.

The acronym RFU is equivalent to Reserved for Future Use.

1.1.3 Prohibited

When a field value is an enumeration, unassigned values can be marked as “Prohibited.” These values shall never be used by an implementation, and any message received that includes a Prohibited value shall be ignored and shall not be processed and shall not be responded to.

Where a field, parameter, or other variable object can take a range of values, and some values are described as “Prohibited,” devices shall not set the object to any of those Prohibited values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous.

“Prohibited” is never abbreviated.

1.2 Requirements in tables

Requirements in this specification are defined as "Mandatory" (M), "Optional" (O), "Excluded" (E), “Not Applicable” (N/A), or "Conditional" (C.#). Conditional statements (C.#) are listed directly below the table in which they appear.

- "M” for mandatory. See Table 1.1.
- "O” for optional. See Table 1.1.

- "E" for excluded. "Excluded" means not permitted in this context; cannot be supported or included for this purpose. The item can still be supported or included if allowed for some other purpose (e.g., a feature can be mandatory for one role and excluded for another; a device that supports both roles must support this feature).
- "C.#" for conditional. "Conditional" means that an item is required, optional, or prohibited based on whether one or more other items are supported or included (# represents any number). Within the definition of the condition, if those other items mean that "not permitted" applies, it has the same meaning as "E".

1.2.1 Table requirements for unused cells

An unused cell is a table cell without a value or content, which is indicated with either the word "none" (without quotation marks) or a hyphen (i.e., a "minus" sign).

1.3 Conformance

Each capability of this specification shall be supported in the specified manner. This specification may provide options for design flexibility, because, for example, some products do not implement every portion of the specification. For each implementation option that is supported, it shall be supported as specified.

1.4 Byte transmission order

All characteristics used with this service shall be transmitted with the least significant octet (LSO) first (i.e., little endian). Where the format is described in tables in this document, the LSO is the first octet in the topmost field of the table; the most significant octet (MSO) is the last octet in the bottommost field of the table. Where characteristics are defined in the GATT Specification Supplement (GSS), see GSS Section 2.2 [1] for more information on octet ordering.

2 Service

2.1 Service dependencies

VAS does not depend on any other service.

2.2 Core Configuration dependencies

Enabling Bluetooth wireless technology end-to-end in an implementation of this service requires an implementation of a Core Configuration that supports the LE transport.

2.2.1 Core feature inter-layer dependencies

There are no additional dependencies beyond minimal compliance with the Core Configuration requirements.

2.3 Dependencies on other specifications external to the Core

There are no dependencies on other specifications external to the Bluetooth Core Specification [2].

2.4 Attribute Protocol Application error codes

VAS and GVAS do not define or reference any Attribute Protocol Application error codes.

2.5 GATT inter-layer dependencies

Dependencies in this section represent a minimum set of capabilities for a Server. Other GATT inter-layer dependencies may be used if supported by both GATT Client and GATT Server.

Table 2.1 summarizes additional GATT inter-layer dependencies beyond those required by all GATT Servers.

GATT Inter-layer Dependency	Support
Notifications (see Volume 3, Part G, Section 4.10 in [2])	M
Read Characteristic Value (see Volume 3, Part G, Section 4.8.1 in [2])	M
Read Long Characteristic Value (see Volume 3, Part G, Section 4.8.3 in [2])	C.1
Write Characteristic Value (see Volume 3, Part G, Section 4.9.3 in [2])	M
Write Without Response (see Volume 3, Part G, Section 4.9.1 in [2])	M
Read Characteristic Descriptor (see Volume 3, Part G, Section 4.12.1 in [2])	M
Write Characteristic Descriptor (see Volume 3, Part G, Section 4.12.3 in [2])	M

Table 2.1: GATT inter-layer dependencies

C.1: Mandatory if the Voice Assistant Name, Installed Location, or Supported Languages characteristics support values that exceed the minimum ATT_MTU - 3 for the Unenhanced ATT bearer, otherwise Optional

2.6 Declaration

This specification describes two services: VAS and GVAS.

VAS shall be a «Primary Service» and the service Universally Unique Identifier (UUID) shall be set to «Voice Assistant Service» as defined in [3]. There may be zero or more instances of VAS.

GVAS shall be a «Primary Service» and the service UUID shall be set to «Generic Voice Assistant Service» as defined in [3]. There shall be one instance of GVAS.

All requirements and descriptions defined for VAS in this specification shall apply to GVAS unless it is explicitly stated otherwise.

2.7 Behavior

VAS is instantiated on devices that support VA functionality. These devices include smartphones, tablets, personal computers (PCs), smartwatches, smart speakers/displays, and conference room equipment.

VAS exposes characteristics that provide status and control of a VA. A device can have multiple, logically separate VAs. A VAS instance provides status and control for a specific VA within the device. A device implements VAS instances when the device wants to allow clients to directly access and control the characteristics of separate VAs.

GVAS provides a single point of access and exposes a representation of the device's default VA. This service provides VA status and control with a single set of characteristics for clients that require a default VA to be set, or for clients that lack a mechanism to select a particular VA. Determining which VA the GVAS instance represents at any given time is left to the implementation.

For example, a smartphone or PC can have many end-user applications that expose VA functionalities. The device can treat these applications as unique VAs. The device can implement a separate VAS instance for each VA, allowing clients to discover, observe, and control the VAs individually. In addition, the device must, as required in Section 2.6, implement a single GVAS instance to provide status and control of the default VA configured on the device.

3 Service characteristics

This section defines the characteristic and descriptor requirements. Where a characteristic can be indicated or notified, a Client Characteristic Configuration Descriptor (CCCD) must be included in that characteristic as required by [2].

Table 3.1 specifies the characteristic and descriptor requirements for each instance of VAS and the single instance of GVAS. These characteristics may be notified, read, or written where the characteristic properties allow.

If the characteristic is configured for notifications and the characteristic value changes while in a connection, then the characteristic shall be notified. Upon reconnection with a bonded client, if a characteristic is configured for notifications and the value of the characteristic changed while the device was disconnected, then the characteristic shall be notified. There shall be only one instance of VAS characteristics within the service definition. The VAS Server shall maintain independent characteristic values for each client.

Characteristic Name	Require-ment	Mandatory Properties	Optional Properties	Security Permissions
Voice Assistant Name (see Section 3.1)	M	Read, Notify	None	Encryption required
Voice Assistant UUID (see Section 3.2)	M	Read, Notify	None	Encryption required
VAS Control Point (see Section 3.3)	M	Write Without Response, Notify	None	Encryption required
Installed Location (see Section 3.4)	O	Read	Notify	Encryption required
Content Control ID (CCID) (see Section 3.5)	M	Read, Notify	None	Encryption required
Voice Assistant Session State (see Section 3.6)	M	Read, Notify	None	Encryption required
Voice Assistant Session Flag (see Section 3.7)	O	Read, Notify	None	Encryption required

Characteristic Name	Requirement	Mandatory Properties	Optional Properties	Security Permissions
Voice Assistant Supported Languages (see Section 3.8)	O	Read, Notify	None	Encryption required
Voice Assistant Supported Features (see Section 3.9)	M	Read, Notify	None	Encryption required

Table 3.1: Requirements for characteristics used in VAS

3.1 Voice Assistant Name characteristic

The Voice Assistant Name characteristic is configured by the VA. The VA vendor is responsible for selecting the value. For example, this characteristic could have the string, “Norman’s Voice Assistant”.

The characteristic is an 8-bit UCS/UTF-8 encoded, human-readable text string.

3.1.1 Voice Assistant Name format

Field Name	Size (Octets)	Format
Voice Assistant Name	Varies	UTF-8

Table 3.2: Voice Assistant Name characteristic value format

3.1.2 Voice Assistant Name behavior

The Voice Assistant Name characteristic returns its associated value when read by a client using the GATT Read Characteristic Value or Read Long Characteristic Value sub-procedure. The client uses the GATT Read Long Characteristic Value sub-procedure if the length of the Voice Assistant Name characteristic exceeds the current ATT_MTU - 3.

The Voice Assistant Name characteristic shall be a non-empty value for each instance of VAS.

3.2 Voice Assistant UUID characteristic

The Voice Assistant UUID characteristic is provided by the VA. The VA vendor is responsible for selecting a UUID value that is consistent across all VAS host platforms and shall use industry standard methods to randomly select a UUID, as specified in the Bluetooth Core Specification (Volume 1, Part E, Section 2.6) [2].

3.2.1 Voice Assistant UUID format

The Voice Assistant UUID characteristic shall be a 128-bit value.

Field Name	Size (Octets)	Format
Voice Assistant UUID	16	uint8

Table 3.3: Voice Assistant UUID characteristic value format

3.3 VAS Control Point characteristic

The VAS Control Point characteristic is used by the client to configure and control the VA.

3.3.1 VAS Control Point format

The VAS Control Point characteristic varies in length, depending on which opcode is used. The first octet is an enumerated value known as the opcode. The parameter fields that follow the opcode have different definitions and purposes based on the opcode. The format and content are described Figure 3.1, Table 3.4, and Table 3.5.

Opcode (1 octet)	Parameter (variable)

Figure 3.1: VAS Control Point characteristic format

Operations Commands			
Opcode Name	**Opcode**	**Para-meter**	**Description**
Initialize VA Session	0x00	None	Client requests to initialize the VA to its initial state.
Start VA Session	0x01	None	Client requests to start a VA session.
Stop VA Session	0x02	None	Client requests to stop a VA session.
RFU	0x03–0xFF	RFU	RFU

Table 3.4: VAS Control Point characteristic opcodes and parameters

Responses		
Opcode Name	**Opcode**	**Parameter**
Response Code	0x00	Response Code Value
RFU	0x01-0xFF	RFU

Table 3.5: VAS Control Point responses

The Response Code Values associated with opcode 0x00 are defined in Table 3.6.

Response Code Value	Definition	Description
0x00	RFU	RFU
0x01	Success	Response for a successful operation
0x02	Opcode Not Supported	Response if an unsupported opcode is received
0x03	Operation Failed	Response for an unsuccessful operation
0x04	Invalid Session State	Response if an opcode is written while the VA is in a Session State that cannot accept it (e.g., Start VA Session Opcode written in the Session Unavailable state)
0x05-0xFF	RFU	RFU

Table 3.6: Response Code Values associated with opcode 0x00

3.3.2 VAS Control Point behavior

The VAS Server shall support all Command Opcodes specified in Table 3.4, and all Response Opcodes specified in Table 3.5. If the CCCD has not been appropriately configured, the server shall ignore the command. If the CCCD has been appropriately configured by the client, then the VAS Server shall notify a Response Opcode after the client writes a Command Opcode to the VAS Control Point characteristic. The Response Code Value will reflect whether the operation was successful or not, according to Table 3.5. Figure 3.2 describes the VA Session States, and the transition conditions between each one.

3.3.2.1 Initialize VA Session

The client can write the Initialize VA Session opcode to the VAS Control Point characteristic in any state except the Session Unavailable state to reset the VA. The value of the Voice Assistant Session State characteristic shall change to Session Ready if the operation is successful. If the client writes this opcode when the Voice Assistant Session State is in the Session Unavailable state, then the VAS Server shall return an error using the Response Code opcode and the Invalid Session State Response Code Value in the parameter. For any other error conditions, the VAS Server shall return an error using the Response Code opcode and the Operation Failed Response Code Value in the parameter.

3.3.2.2 Start VA Session

The client can write the Start VA Session opcode to the VAS Control Point in the Session Ready state to initiate a VA Session. The value of the Voice Assistant Session State characteristic shall change to Session Active if the operation is successful. If the client writes this opcode when the Voice Assistant Session State is in a different state, then the VAS Server shall return an error using the Response Code opcode and the Invalid Session State Response Code Value in the parameter. For any other error conditions, the VAS Server shall return an error using the Response Code opcode and the Operation Failed Response Code Value in the parameter.

3.3.2.3 Stop VA Session

The client can write the Stop VA Session opcode (0x02) to the VAS Control Point in the Session Active state to terminate a VA Session. The value of the Voice Assistant Session State characteristic shall change to Session Ready when the operation succeeds. If the client writes this opcode when the Voice Assistant Session State is

in a different state, then the VAS Server shall return an error using the Response Code opcode and the Invalid Session State Response Code Value in the parameter. Otherwise, if the operation results in an error condition, then the VAS Server shall return an error using the Response Code opcode and the Operation Failed Response Code Value in the parameter.

3.4 Installed Location characteristic

The Installed Location characteristic is configured by the user and read by the client. For example, this characteristic could have the string, “Bedroom” or “Kitchen”. The method for the VAS Server to provide a human-readable service name for the user is outside the scope of this specification. If an Installed Location characteristic is not configured, then the VAS Server shall initialize the Installed Location characteristic to an empty value.

3.4.1 Installed Location format

Field Name	Size (Octets)	Format
Installed Location	Variable	UTF-8

Table 3.7: Voice Assistant Service Name characteristic value format

3.4.2 Installed Location behavior

The Installed Location characteristic returns its associated value when read by a client using the GATT Read Characteristic Value or Read Long Characteristic Value sub-procedure and is supported by the VAS Server.

3.5 Content Control ID characteristic

3.5.1 Content Control ID format

The Content Control ID (CCID) characteristic is defined in [1].

3.5.2 Content Control ID behavior

The CCID characteristic returns its associated value when read by a client using the GATT Read Characteristic Value sub-procedure.

The CCID characteristic in each instance of the service shall persist across connections until the handle range is affected by a service change.

3.6 Voice Assistant Session State characteristic

The Voice Assistant Session State characteristic exposes the current VA Session State that is maintained by the VAS Server (see Table 3.9).

3.6.1 Voice Assistant Session State format

The Voice Assistant Session State characteristic conveys the Session State of the VA.

The characteristic consists of one octet with the format shown in Table 3.8.

Field Name	Size (Octets)	Format
VA Session State	1 octet	uint8

Table 3.8: Voice Assistant Session State characteristic format

The values of the Session State field are defined in Table 3.9.

Session State	Value	Description
Session Reset	0x00	When the VA is reset but not initialized and not ready, typically after a connection is established.
Session Unavailable	0x01	When the VA is not available.
Session Ready	0x02	When the VA has been initialized by the client and is now ready for the Start VA Session and Stop VA Session VAS Control Point operations.
Session Active	0x03	When the VA has entered a state of engaging with the user.
RFU	0x04-0xFF	RFU

Table 3.9: Voice Assistant Session State characteristic Session State values

3.6.2 Voice Assistant Session State behavior

The Voice Assistant Session State characteristic returns the value associated with the VA Session State when read by a client using the GATT Read Characteristic Value sub-procedure.

The initial VA Session State upon connection shall be set to Session Reset.

Figure 3.2 describes the VA Session States, and the transition conditions between each one of them.

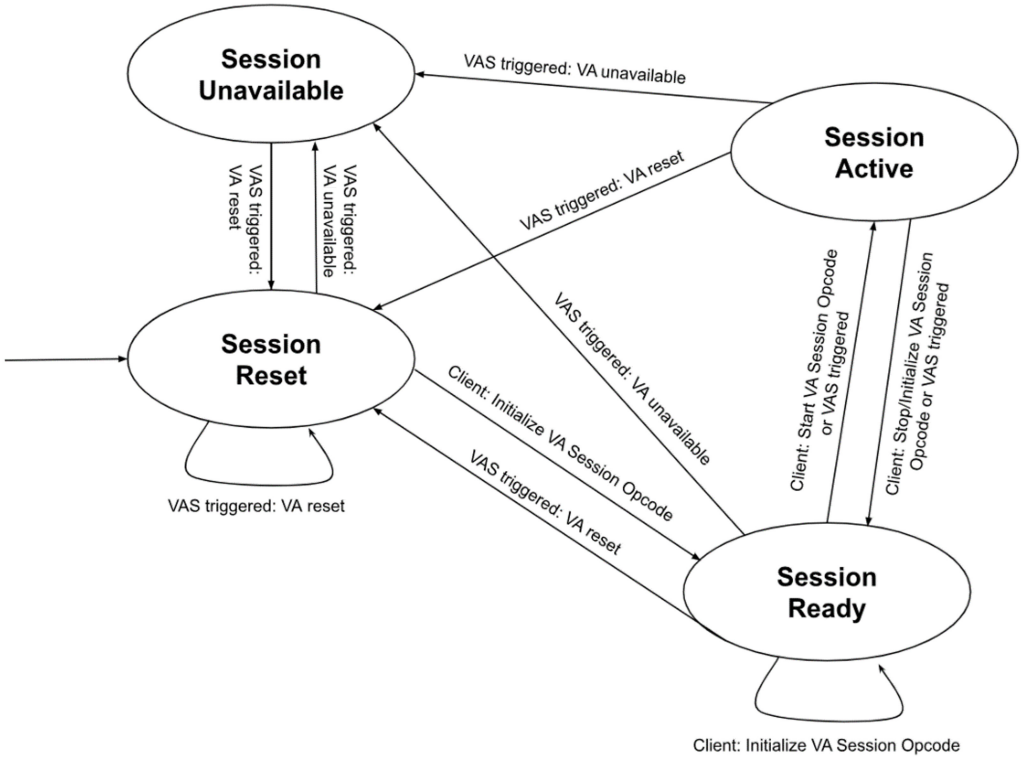

Figure 3.2: VA Session State transitions

3.6.2.1 Session Unavailable

If the VAS Server has no valid VA attached, or the VA has determined that it is unable to provide service (e.g., no Cellular or Wi-Fi service), then the VAS Server shall be in the Session Unavailable state. If any of these conditions change, then the VAS Server shall transition to the Session Reset state.

VAS Server can transition to the Session Unavailable state from any other state unilaterally.

3.6.2.2 Session Reset

The VAS Server shall be in the Session Reset state upon power up or reconnection. The GVAS Server shall transition to the Session Reset state when the default VA is changed. The VAS Server should transition to the Session Reset state if the VA encounters a critical error or is reset.

The VAS Server may also transition from any state to the Session Reset state unilaterally.

When transitioning to the Session Reset state, the VAS Server may update the values of all the characteristics with the Read permission (e.g., the location changes, the VA name updates after a software update). This enables the client to read the values of the characteristics it is interested in if the client becomes aware that the VA Session State value is set to Session Reset.

3.6.2.3 Session Ready

The VAS Server shall transition from the Session Reset or Session Active states to the Session Ready state when an Initialize VA Session opcode write operation by the client is completed.

The VAS Server shall also transition from the Session Active state to the Session Ready state when a Stop VA Session opcode write operation by the client is completed.

The VAS Server may also transition from the Session Active state to the Session Ready state unilaterally.

3.6.2.4 Session Active

The VAS Server shall transition from the Session Ready state to the Session Active state when a Start VA Session opcode write operation by the client is completed.

The VAS Server may also transition from the Session Ready state to the Session Active state unilaterally.

3.7 Voice Assistant Session Flag characteristic

The Voice Assistant Session Flag characteristic conveys additional details about the VA Session State when the Session Flags Enabled is set in the Voice Assistant Supported Features characteristic (see Section 3.9) and the VA is in the Session Active state.

3.7.1 Voice Assistant Session Flag format

The characteristic consists of 1 octet with the format shown in Table 3.10.

Field Name	Size (Octets)	Format
VA Session Flag	1 octet	uint8

Table 3.10: Voice Assistant Session Flag characteristic format

The bit field values are defined in Table 3.11.

Flag Name	Bit	Description
Listening Now	Bit 0	0 = The VA is not listening 1 = The VA is listening
Processing Now	Bit 1	0 = The VA is not processing any request 1 = The VA is processing request
Playback Now	Bit 2	0 = The VA is silent 1 = The VA is responding
RFU	Bit 3-7	RFU

Table 3.11: Voice Assistant Session Flag characteristic Session Flag values

3.7.2 Voice Assistant Session Flag behavior

The Voice Assistant Session Flag characteristic returns its associated value when read by a client using the GATT Read Characteristic Value sub-procedure. The values of Listening Now, Processing Now, and Playback Now bit fields can be set when the Session Flags Enabled is set in the Voice Assistant Supported Features characteristic and the VA Session State is in the Session Active state. In any other VA Session States, the VA Session Flags value shall be set to 0.

3.7.2.1 Listening Now

The purpose of the Listening Now bit field is as follows: The VA sets the Listening Now bit to indicate when the VA is listening for voice commands or instructions. The VA clears this bit when it is not listening.

3.7.2.2 Processing Now

The purpose of the Processing Now bit field is as follows: The VA sets the Processing Now bit to indicate when the VA is busy, for example, when the user has completed giving the VA an instruction and the VA is processing that instruction or is carrying out that instruction. The VA clears the Processing Now bit when the VA is no longer processing a command or instruction. A client may have the ability to display a “Processing Now” status to the user; displaying this status makes the user aware the VA is processing their request.

3.7.2.3 Playback Now

The purpose of the Playback Now bit field is to indicate that the VA is actively responding to the user’s request. The VA sets the Playback Now bit value immediately prior to responding, and clears the Playback Now bit immediately after completing the response.

3.8 Voice Assistant Supported Languages characteristic

The Voice Assistant Support Languages characteristic is provided by the VA to allow the client to discover the spoken languages that are recognized by the VA.

3.8.1 Voice Assistant Supported Languages format

The characteristic is a UTF-8 encoded string. The format is a comma-separated list of languages, each element in the list is an ISO 639-3 value (see [4]).

3.8.2 Voice Assistant Supported Languages behavior

The Voice Assistant Support Languages characteristic returns its associated value when read by a client using the GATT Read Characteristic Value or Read Long Characteristic Value sub-procedure and is supported by the VAS Server.

3.9 Voice Assistant Supported Features characteristic

The Voice Assistant Supported Features characteristic is a bitmap that indicates the capabilities and features supported by the VA.

3.9.1 Voice Assistant Supported Features format

The characteristic consists of 1 octet with the format shown in Table 3.12.

Field Name	Size (Octets)	Format
VA Supported Features	1 octet	uint8

Table 3.12: Voice Assistant Supported Features characteristic format

The bit field values are defined in Table 3.13.

Flag Name	Bit	Description
Session Flags Enabled	Bit 0	0 = VA Session flag values are not enabled (i.e., they do not convey valid information about the VA Session) 1 = VA Session flag values are enabled (i.e., they convey valid information about the VA Session)
RFU	Bit 1-7	RFU

Table 3.13: Voice Assistant Supported Features characteristic values

3.9.2 Voice Assistant Supported Features behavior

The Voice Assistant Supported Features characteristic returns its associated value when read by a client using the GATT Read Characteristic Value sub-procedure.

3.9.2.1 Session Flags Enabled

The Session Flags Enabled bit indicates to the client that the Voice Assistant Session Flag characteristic is enabled, i.e., the bit fields are valid. The Voice Assistant Session Flag characteristic shall not be notified or changed when the Session Flags Enabled bit is not set.

4 Acronyms and abbreviations

Acronym/Abbreviation	Meaning
ACL	Asynchronous Connection-oriented logical
ATT	Attribute Protocol
CCCD	Client Characteristic Configuration Descriptor
CCID	Content Control Identifier
GATT	Generic Attribute Profile
GVAS	Generic Voice Assistant Service
LE	Low Energy
RFU	Reserved for Future Use
UTF-8	8-bit UCS/Unicode Transformation Format
UUID	Universally Unique Identifier
VA	Voice Assistant
VAS	Voice Assistant Service

Table 4.1: Acronyms and abbreviations

5 References

[1] GATT Specification Supplement, https://www.bluetooth.com/specifications/gss/

[2] Bluetooth Core Specification, Version 6.1 or later

[3] Bluetooth Assigned Numbers, https://www.bluetooth.com/specifications/assigned-numbers/

[4] ISO 639-3:2007: Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages, https://iso639-3.sil.org/code_tables/639/data