Low Energy Audio – Using UCIs, URIs, and URI Schemes

***Bluetooth*® Informational Publication**

- **Version:** v1
- **Version Date:** 2024-06-11
- **Prepared By:** Generic Audio Working Group

If you have any questions or comments on this informational publication or Bluetooth LE Audio specifications, please submit a ticket: https://www.bluetooth.com/leaudio-spec-feedback/

Abstract:

Manufacturers sometimes create Bluetooth products with calling features beyond simply answering or ending a call. Features might include initiating a call to a contact from a list; selecting a specific VoIP application to use for a call; or manufacturing or configuring a headset or similar device to work with a specific application.

This document describes how implementers can use the UCIs, URIs, and URI schemes in the Call Control Profile (CCP) [1] and the Telephone Bearer Service (TBS) [2] specifications to develop these product features. Implementers will learn how to visualize and understand the relationships between UCIs, URIs, and URI schemes, and how to integrate them into features.

Version History

Version Number	Date (yyyy-mm-dd)	Comments
v1	2024-06-11	Approved by the Bluetooth SIG Board of Directors.

Acknowledgments

Name	Company
Frank Yerrace	Microsoft Corporation
Nick Hunn	GN Hearing A/S

Contents

1 Introduction

Some features of the Call Control Profile (CCP) [1] and the Telephone Bearer Service (TBS) [2] specifications use Uniform Caller Identifiers (UCIs), Uniform Resource Identifiers (URIs), and URI schemes. These allow headsets and similar devices to discover and interact with specific Voice over Internet Protocol (VoIP) or calling applications commonly found on smartphones and personal computers; to identify callers; to initiate a call to a contact; and to perform other functions.

This document helps implementers understand how to use these specification features in their product designs.

This is not a Bluetooth specification, therefore, the established Bluetooth SIG specification language conventions for use of the words ***shall***, ***shall not***, ***must***, ***should***, ***should not***, ***may***, and ***can*** do not apply to this document.

1.1 Scope

This document:

- Shows a simple but practical example of a Call Control Server device and a Call Control Client device.
- Describes TBS and the Generic Telephone Bearer Service (GTBS) [2].
- Uses an example to explain how UCIs, URIs, and URI schemes are used in these services.

2 Example

Figure 2.1 shows examples of a Call Control Server and a Call Control Client that illustrate the differences and relationships between UCIs, URIs, and URI schemes.

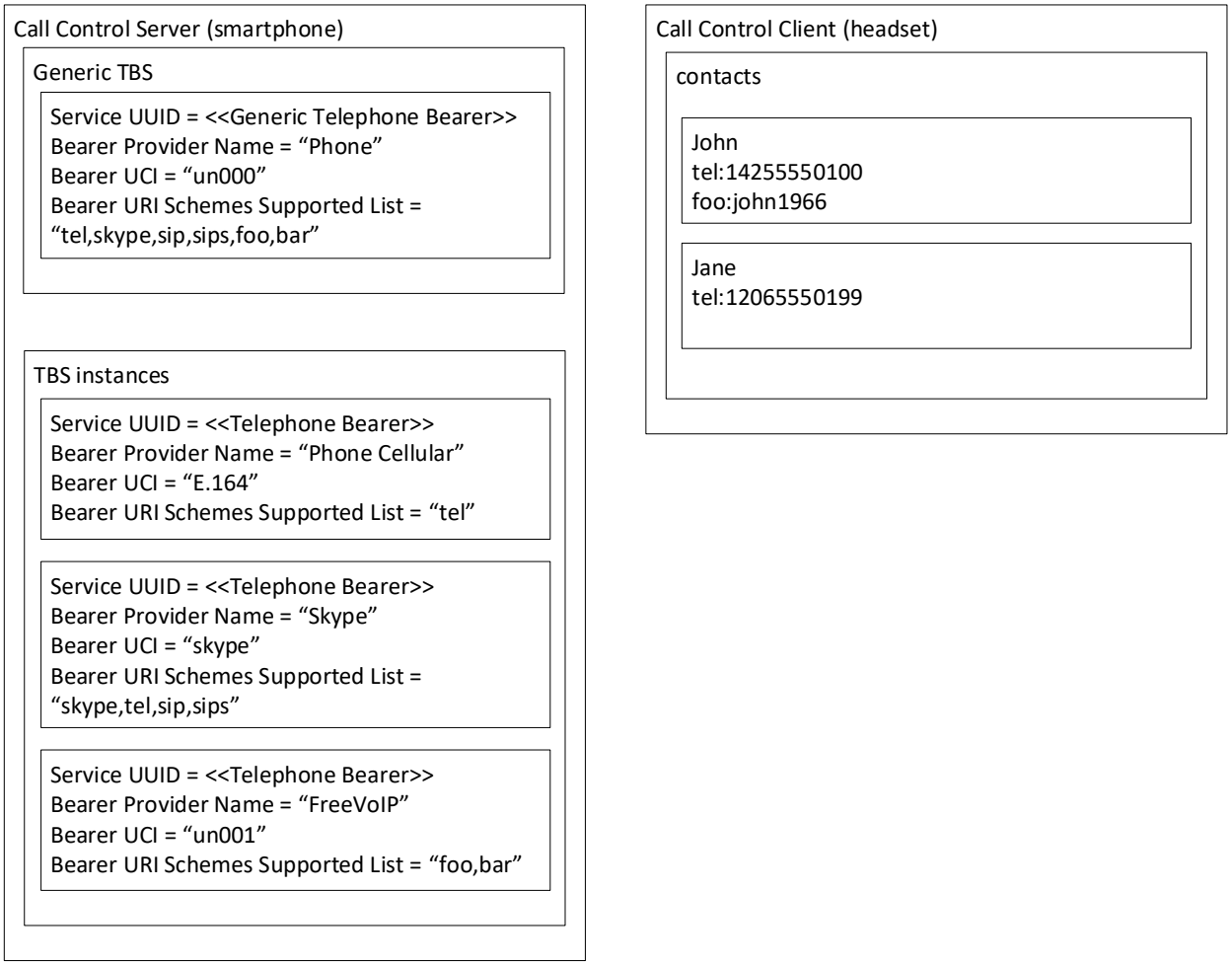

Figure 2.1: Example Call Control Server and Call Control Client

In this example, the Call Control Server device is a smartphone and the Call Control Client device is a headset. The headset has two contacts stored, John and Jane, obtained through a mechanism outside the scope of this paper, such as the Phone Book Access Profile.

The smartphone exposes several instances of TBS [2] that represent the smartphone's cellular calling function and installed VoIP applications. The headset can use the different TBS instances to interact with specific applications (described in Section 4).

The smartphone also exposes a single GTBS [2] that aggregates the capabilities of the VoIP applications and cellular calling. The smartphone implementer can choose which calling capabilities are available through GTBS; the smartphone manufacturer should consider that some Call Control Client devices might use only GTBS to avoid the complexity of discovering and selecting among TBS instances.

UCIs, URIs, and URI schemes are used as follows:

- A UCI is a persistent identifier of a specific VoIP or calling application. In the example in Figure 2.1, the UCI “skype” uniquely identifies the TBS instance associated with the Skype application.
- A URI[1] identifies a contact. In the example in Figure 2.1, the URI “tel:14255550100” identifies the contact John. The URI “foo:john1966” also identifies John.
- A URI scheme defines the format of the contact identifier data. Each TBS or GTBS can express its supported URI schemes. In the example in Figure 2.1, the (fictitious) FreeVoIP application supports the (fictitious) foo and bar URI schemes.

Some URI schemes suggest that a particular application will handle that URI. For example, the skype URI scheme is usually handled by the Skype application. However, multiple applications, each identified by a unique UCI, might support and potentially handle a given URI scheme. In the example in Figure 2.1, both Skype and Phone Cellular applications can handle the tel URI scheme. Similarly, a single application might handle many different URI schemes. For example, the Skype application can handle skype and tel URI schemes, as well as several others.

[1] Although the term URI has a more general definition in the computing industry, in the context of CCP [1], it is used only to identify a contact.

3 Binding UCIs to applications

UCIs are published in the Uniform Caller Identifiers section of Bluetooth Assigned Numbers [3] and identify specific user applications.

3.1 Known applications

The Call Control Server binds a UCI from Bluetooth Assigned Numbers to the application when it identifies the application and instantiates a TBS associated with the application. The Call Control Server in the example in Figure 2.1 binds the UCI “skype” to the Skype application, and the UCI “E.164” for calling through a smartphone’s mobile service provider.

3.2 Unknown applications

If the Call Control Server determines that the application has no UCI defined in Assigned Numbers, or cannot identify the application (for example, because of implementation constraints), then the Call Control Server assigns a UCI from the Unknown range defined in Bluetooth Assigned Numbers having the form “un000” to “un999”, selecting a UCI that it has not assigned to another application. These values from the Unknown range are limited in scope to the Call Control Server device; other Call Control Server devices might bind these UCIs to different applications.

The example Call Control Server in Figure 2.1 binds the UCI “un001” to the FreeVoIP application because that application does not have a UCI assigned in Bluetooth Assigned Numbers. Another Call Control Server device might bind the “un001” UCI to a different application. Despite the existence of the “skype” UCI in Bluetooth Assigned Numbers, the example Call Control Server device could have bound a UCI like “un002” to the Skype application if the device was unable to determine that the application is Skype.

Call Control Client devices might store the UCIs read from TBS instances in order to interact with the same Call Control Server device applications again in the future. Therefore, when a Call Control Server device binds a UCI from the Unknown range “un000” to “un999”, it is expected that the same UCI is assigned across application restart, device restart, and even application reinstall. In the example in Figure 2.1, the Call Control Server should assign the same UCI “un001” to the FreeVoIP application after the user either restarts the FreeVoIP application or restarts the device.

However, implementations can have practical limitations that make persistence across certain internal events difficult or impossible, causing the UCI for a given application to change. For example, if the Call Control Server device restarts and cannot determine that “un001” had been previously assigned to FreeVoIP, the Call Control Server device might assign a different UCI from the Unknown range and create a different TBS instance for FreeVoIP. Call Control Client devices will interpret the new UCI and TBS instance as a different unknown application.

In the example of Figure 2.1, the Call Control Server should assign the same UCI “un001” to the FreeVoIP application after the user either restarts the FreeVoIP application or restarts the device.

When assigning UCIs from the Unknown range, the Call Control Server device should avoid previously used UCIs that might have been assigned to a different application in the past. For example, the Call Control Server device can generate UCIs from the Unknown range using a simple counter that tracks the last-used UCI and increment the counter to generate the next UCI. To avoid reusing UCIs from the Unknown range across reboots, the device can store the counter or generate a random starting counter value after reboot.

3.3 UCIs for GTBS

The proper UCI for GTBS depends on the Call Control Server implementation. If the Call Control Server's GTBS aggregates and interacts with several calling applications, the Call Control Server will have a UCI from the Unknown range. The Call Control Server in the example in Figure 2.1 binds "un000". If the Call Control Server's GTBS interacts with only one application (e.g., only cellular calls), the Call Control Server's GTBS can have the UCI that identifies that application.

4 Using UCIs in a Call Control Client device

Many Call Control Client devices do not differentiate between VoIP or calling applications and interact with the GTBS [2] of a Call Control Server device. However, a Call Control Client device can use a UCI to interact with a specific application through a TBS instance.

Two example scenarios are:

- Case 1, where the user has selected an application from some UI on the Call Control Client device.
- Case 2, where the Call Control Client device was manufactured to use or prefer a specific application.

In Case 1, the Call Control Client device can discover all the TBS instances of the Call Control Server device and read each Bearer UCI and Bearer Provider Name to display a list of applications to the user. The Call Control Client device would not interpret or attach any specific meaning to the Bearer UCI; it only uses the UCI as a persistent application identifier, which it stores after the user selects an application from the list.

To interact with the same application later, the Call Control Client device can search the discovered TBS instances to find one that has a Bearer UCI matching the stored value, and then interact with that TBS instance to place a call. In this case, the Call Control Client device is not concerned with whether UCIs are assigned from the Unknown range. However, persistence is important so that the Call Control Client device can interact with the same application after either the application or the Call Control Server device restarts.

Case 2 is similar to Case 1 except that, instead of getting the UCI from a user's selection, the UCI is coded into the Call Control Client device during manufacture; the Call Control Client device is manufactured to prefer a particular application. In this case, the UCI does need to be a preassigned UCI, not from the Unknown range.

As described above, UCIs can be used by a Call Control Client to find a specific instance of TBS, However, UCIs are not useful for finding the GTBS instance, as there is only one instance of GTBS and it is easily discovered using its service UUID.

5 Summary and conclusions

Using the UCI, URI, and URI schemes features of CCP and TBS, devices can be manufactured with more advanced calling features beyond answering and ending calls such as initiating a call to a contact from a list; selecting a specific VoIP application to use for a call; or manufacturing or configuring a headset or similar device to work with a specific application.

6 References

[1] Call Control Profile Specification, Version 1.0

[2] Telephone Bearer Service Specification, Version 1.0

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