

# Calendar Tasks and Notes Profile (CTN)

*Bluetooth*<sup>®</sup> Profile Specification



- **Date** 2014-Sep-18
- **Revision** V1.0.0
- **Group Prepared By** Telephony and Car Working Group
- **Feedback Email** [car-main@bluetooth.org](mailto:car-main@bluetooth.org)

## **Abstract:**

This Profile Specification defines a set of features and procedures to exchange calendar, task and note objects between devices. It is especially tailored for use cases where a client device (e.g., a car-kit or another terminal device with IO-capabilities) accesses the CTN-object repository of a server device (typically a mobile phone or a smartphone). However, this profile can also be used for other use cases that require the exchange of CTN objects between two devices.

### Revision History

Revision Number	Date	Comments
V1.0.0	18-09-2014	Adopted by the Bluetooth SIG BoD

### Contributors

Name	Company
Veit Kötting	Berner&Mattner
Joachim Mertz	Berner&Mattner
Dominik Sollfrank	Berner&Mattner
Victor Zhodzishsky	Broadcom
Burch Seymour	Continental Automotive Systems
Doron Elliott	Ford
Anil Vutukuru	Mindtree
Kevin Hendrix	OpenSynergy
Kyle Penri-Williams	Parrot
David Hughes	Broadcom
Casper Bonde	Samsung
Kim Schulz	Samsung
Trine Borgbjerg Jensen	Samsung

V1.0.0

**DISCLAIMER AND COPYRIGHT NOTICE**

This disclaimer applies to all draft specifications and final specifications adopted by the Bluetooth SIG Board of Directors (both of which are hereinafter referred to herein as a Bluetooth "Specification"). Your use of this Specification in any way is subject to your compliance with all conditions of such use, and your acceptance of all disclaimers and limitations as to such use, contained in this Specification. Any user of this Specification is advised to seek appropriate legal, engineering or other professional advice regarding the use, interpretation or effect of this Specification on any matters discussed in this Specification.

Use of Bluetooth Specifications and any related intellectual property 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, including, but not limited to, the Membership Application, the Bluetooth Patent/Copyright License Agreement and the Bluetooth Trademark License Agreement (collectively, 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 use of any portion of a Bluetooth Specification may involve the use of intellectual property rights ("IPR"), including pending or issued patents, or copyrights or other rights. Bluetooth SIG has made no search or investigation for such rights and disclaims any undertaking or duty to do so. The legal rights and obligations of each Member are governed by the applicable Membership Agreements, 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 Agreements, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in (i) termination of the applicable Membership Agreements or Early Adopters Agreement and (ii) liability claims by Bluetooth SIG or any of its Members for patent, copyright and/or trademark infringement claims permitted by the applicable agreement or by applicable law.

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 wireless technology ("Bluetooth Products") may be subject to various regulatory controls under the laws and regulations applicable to products using wireless non licensed spectrum 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. To the extent not prohibited by law, in no event will Bluetooth SIG or its Members or their affiliates be liable for any damages, including without limitation, lost revenue, profits, data or programs, or business interruption, or for special, indirect, consequential, incidental or punitive damages, however caused and regardless of the theory of liability, arising out of or related to any furnishing, practicing, modifying, use or the performance or implementation of the contents of this Specification, even if Bluetooth SIG or its Members or their affiliates have been advised of the possibility of such damages. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS MEMBERS OR THEIR AFFILIATES RELATED TO USE OF THE SPECIFICATION.

If this Specification is an intermediate draft, it is for comment only. No products should be designed based on it except solely to verify the prototyping specification at SIG sponsored IOP events and it does not represent any commitment to release or implement any portion of the intermediate draft, which may be withdrawn, modified, or replaced at any time in the adopted Specification.

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

**Copyright © 2014. The Bluetooth word mark and logos are owned by Bluetooth SIG, Inc. All copyrights in the Bluetooth Specifications themselves are owned by Ericsson AB, Lenovo (Singapore) Pte. Ltd., Intel Corporation, Microsoft Corporation, Motorola Mobility, LLC, Nokia Corporation and Toshiba Corporation. Other third-party brands and names are the property of their respective owners.**

## 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.

## Contents

<b>1</b>	<b>Introduction .....</b>	<b>8</b>
1.1	Scope.....	8
1.2	Conformance .....	8
1.3	Profile Dependencies.....	8
1.4	Bluetooth Specification Release Compatibility .....	9
1.5	Symbols, Conventions, and Definitions .....	9
<b>2</b>	<b>Profile Overview .....</b>	<b>11</b>
2.1	Protocol Stack.....	11
2.2	Configurations, Roles, and Modes .....	11
2.3	User Scenarios .....	12
2.4	Safety Considerations.....	13
<b>3</b>	<b>Application Layer .....</b>	<b>14</b>
3.1	Overview .....	14
3.2	General Attributes and Conventions.....	14
3.2.1	Character Set.....	14
3.2.2	Timestamps .....	14
3.2.3	Handles and Sessions.....	14
3.3	CTN Application Objects .....	15
3.3.1	bCalendar Objects (x-bt/Calendar).....	15
3.3.2	CTN-Listing Object (x-bt/CTN-Listing).....	18
3.3.3	CTN-Event-Report Object (x-bt/CTN-EventReport) .....	22
3.4	CTN Folder-Structure .....	25
3.5	CTN Access Server (CAS) Accounts .....	26
<b>4</b>	<b>CTN Profile Features .....</b>	<b>27</b>
4.1	Overview.....	27
4.2	CTN Account Management Feature.....	27
4.3	CTN Notification Feature .....	28
4.4	CTN Browsing Feature .....	29
4.5	CTN Downloading Feature .....	30
4.6	CTN Uploading Feature.....	31
4.7	CTN Forward Feature .....	32
4.8	CTN Delete Feature.....	33
4.9	Initialization sequences .....	34
<b>5</b>	<b>CTN Profile Functions .....</b>	<b>35</b>
5.1	SendCTNEvent function .....	35
5.1.1	Connection ID .....	35
5.1.2	Type.....	36
5.1.3	Application parameters.....	36

5.1.4	Body/EndOfBody .....	36
5.2	CTNNotificationRegistration function.....	36
5.2.1	Connection ID .....	37
5.2.2	Type .....	37
5.2.3	Application parameters .....	37
5.2.4	Body/EndOfBody .....	37
5.3	GetCTNListing function .....	37
5.3.1	Connection ID .....	38
5.3.2	Type .....	38
5.3.3	Name .....	39
5.3.4	Application parameters .....	39
5.3.5	Body/EndOfBody .....	40
5.4	GetCTNObject function .....	41
5.4.1	Connection ID .....	41
5.4.2	Type .....	41
5.4.3	Name .....	42
5.4.4	Application parameters .....	42
5.4.5	Body/EndOfBody .....	42
5.5	SetCTNStatus function .....	42
5.5.1	Connection ID .....	43
5.5.2	Name .....	43
5.5.3	Type .....	43
5.5.4	Application Parameters .....	43
5.5.5	Body/EndOfBody .....	45
5.6	PushCTNObject function .....	45
5.6.1	Connection ID .....	46
5.6.2	Name .....	46
5.6.3	Type .....	46
5.6.4	Application parameters .....	46
5.6.5	Body/EndOfBody .....	46
5.7	ForwardCTNObject function .....	46
5.7.1	Connection ID .....	47
5.7.2	Name .....	47
5.7.3	Type .....	47
5.7.4	Description .....	47
5.7.5	Body/EndOfBody .....	47
5.8	GetCTNAccountInformation function .....	47
5.8.1	Connection ID .....	48
5.8.2	Type .....	48



5.8.3	Body/End OfBody .....	49
5.8.4	Application parameters .....	49
5.9	SyncCTNAccount function .....	49
5.9.1	Connection ID .....	50
5.9.2	Type .....	50
5.9.3	Body/EndOfBody .....	50
<b>6</b>	<b>Profile Dependencies .....</b>	<b>51</b>
6.1	Profile fundamentals .....	51
6.2	OBEX .....	51
6.2.1	OBEX Services .....	51
6.2.2	OBEX Operations Used .....	51
6.2.3	OBEX Headers .....	52
6.2.4	Application Parameters Header .....	53
6.2.5	OBEX headers in Multi-Packet Responses .....	54
6.2.6	OBEX Error Codes .....	54
6.3	Link Control (LC) Interoperability Requirements .....	54
6.4	Generic Access Profile .....	54
6.5	Modes .....	54
6.6	Security aspects .....	54
6.7	Idle mode procedures .....	54
<b>7</b>	<b>Service Discovery .....</b>	<b>56</b>
7.1	SDP Interoperability Requirements .....	56
7.2	SDP record for the CTN Access Service on the CSE device .....	56
7.3	SDP record for the CTN Notification Service on the CCE device .....	57
<b>8</b>	<b>Acronyms and Abbreviations .....</b>	<b>59</b>
<b>9</b>	<b>References .....</b>	<b>60</b>

# 1 Introduction

## 1.1 Scope

The Calendar, Tasks and Notes Profile (CTN) specifies the exchange of Personal Information Management (PIM) objects between Bluetooth devices. It is based on a Client-Server interaction model where the Client initiates the transactions.

The profile is based on the Generic PIM Profile (GPP). GPP is an abstract profile that defines common requirements for application profiles, in particular

- requirements and constraints concerning lower Bluetooth layers such as Bluetooth Security, GAP requirements, Link Manager/Control and common PIM mechanisms and capabilities
- templates for commonly used functions
- templates for application objects

CTN inherits these general GPP requirements, objects and functions and adds appropriate details. Furthermore, CTN adds application objects, features and functions, descriptive parts and further or refined requirements for the specific CTN use cases.

CTN makes use of the iCalendar standard [8] by embedding iCalendar objects into Bluetooth application objects (see bCalendar definition in Section 3.3.1) and defining a suitable set of operations on them.

## 1.2 Conformance

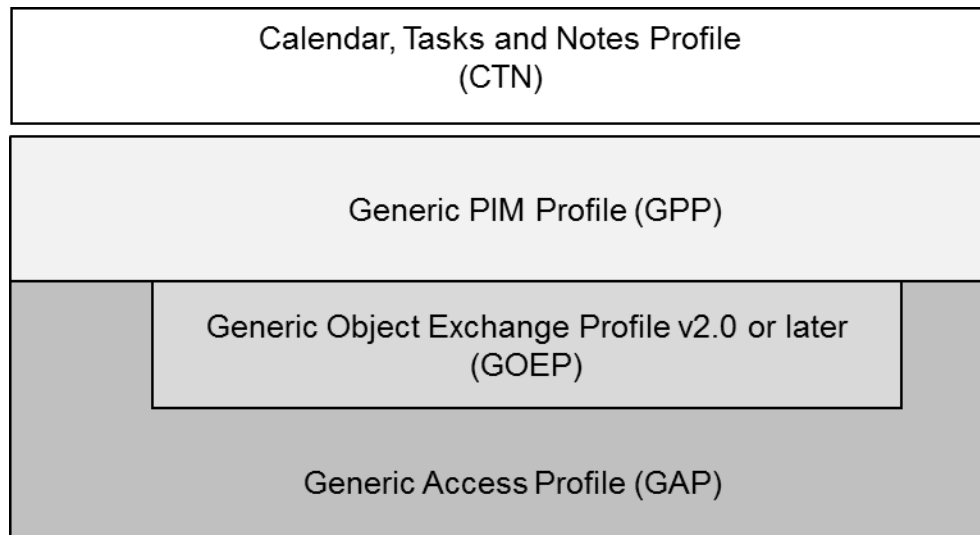
If conformance to this Profile is claimed, all capabilities indicated as mandatory for this Profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional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

## 1.3 Profile Dependencies

Interoperability between devices from different manufacturers is provided for a specific service and usage model if the devices conform to a Bluetooth SIG defined profile specification. A profile defines a selection of messages and procedures (generally termed capabilities) from the Bluetooth SIG specifications and gives an unambiguous description of the air interface for specified service(s) and usage model(s).

A profile is dependent upon another profile if it re-uses parts of that profile by explicitly referencing it. The Bluetooth profile structure and the dependencies of GPP are depicted in the figure below. A profile has dependencies on the lower-level profile(s).





**Figure 1.1:** Dependencies Bluetooth Profiles

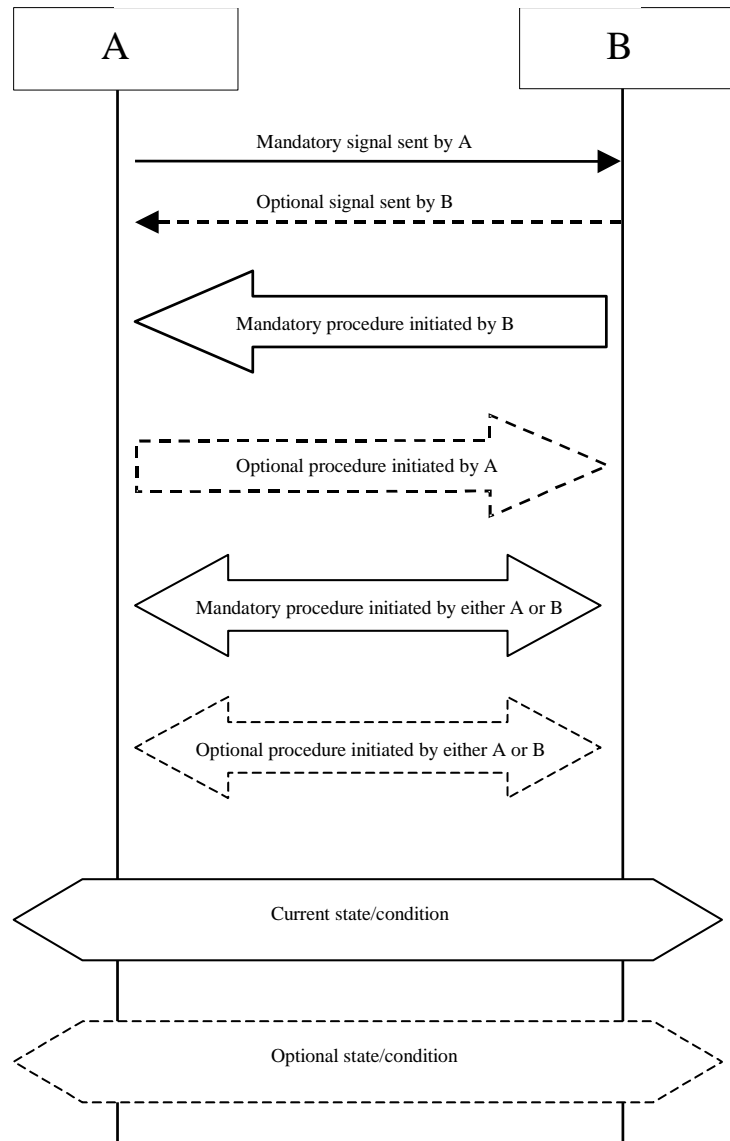
As indicated in the figure, the Calendar, Tasks and Notes Profile is dependent upon the Generic PIM Profile (GPP) [6], the Generic Object Exchange Profile (GOEP) [4] and the Generic Access Profile (GAP) [5]. In particular, GPP and therefore CTN is based on GOEP v2.0 (or later) with OBEX over L2CAP.

## 1.4 Bluetooth Specification Release Compatibility

According to GPP [6], this profile is compatible with the Bluetooth Core specification version 2.1 + EDR or later. For further details see [6].

## 1.5 Symbols, Conventions, and Definitions

The signaling diagrams in this specification are informative only. Within the diagrams, the following conventions are used to describe procedures:

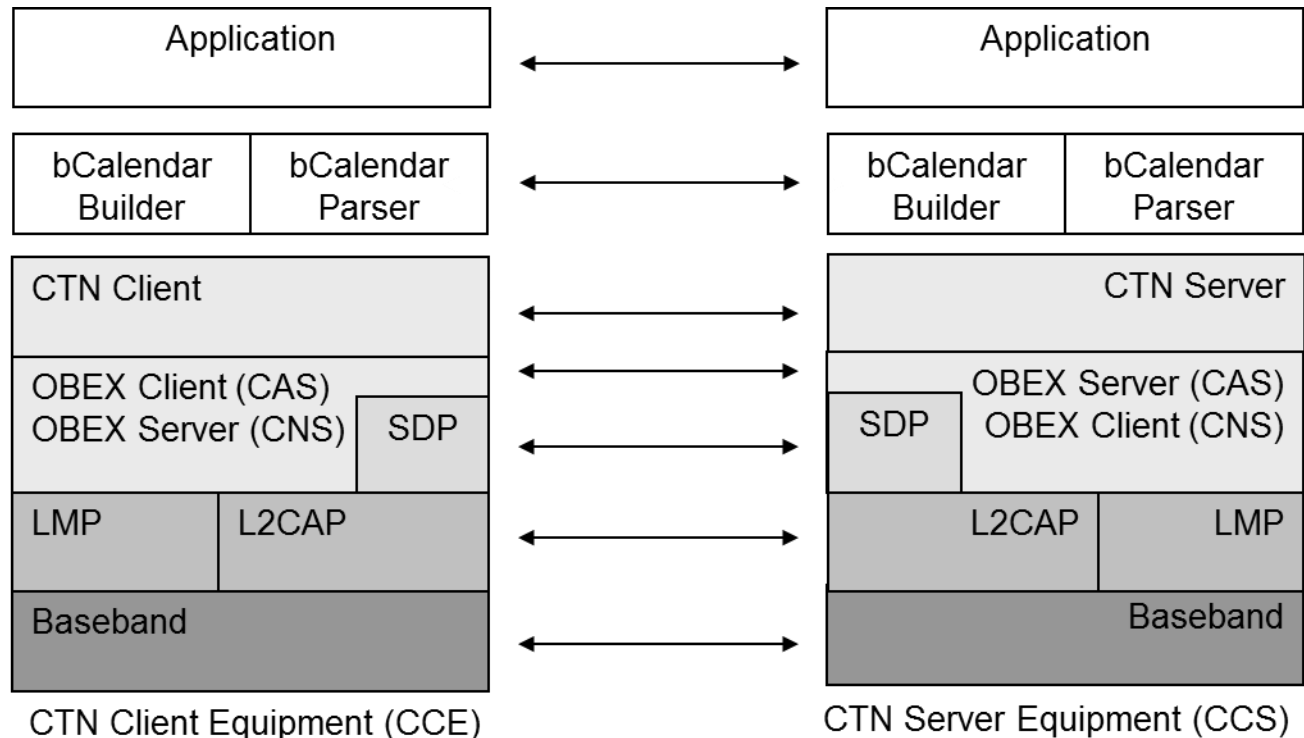


**Figure 1.2:** Conventions used in signaling diagrams

## 2 Profile Overview

### 2.1 Protocol Stack

The figure below shows the protocols and entities used in this profile:



**Figure 2.1:** Protocol model for transport of CTN data objects

CAS = CTN Access Service, CNS = CTN Notification Service, see Section 4.

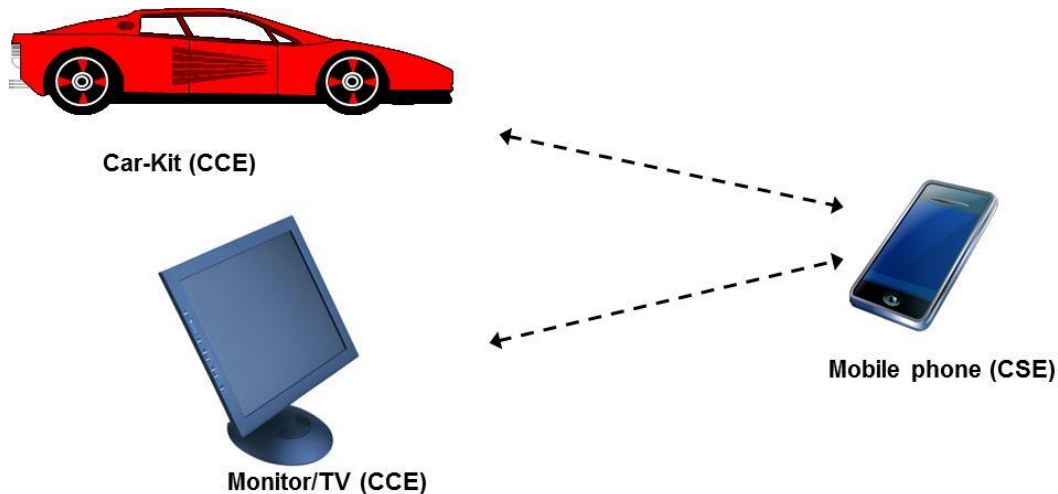
### 2.2 Configurations, Roles, and Modes

The following roles are defined for this profile:

- **CTN Server Equipment (CSE)** – is the device that provides the CTN object repository (i.e., the CSE has the ability to provide a client unit with calendar, task, and note objects that are stored in this device and with notifications of changes in its repository). Furthermore the CSE provides features to a client (e.g., to upload and modify CTN entries in its repository). A CSE device may be a mobile phone or a smartphone.
- **CTN Client Equipment (CCE)** – is the device that accesses the repository engine of the CSE for downloading, browsing and displaying existing CTN objects, modifying CTN objects, or to upload objects to the CSE. A CCE device may be a car head unit.

The terms CSE and CCE are used in the rest of this document to designate these roles.

The figure below shows typical configurations of devices for which the CTN Profile is applicable. Nevertheless, any Bluetooth device with a CTN object repository can potentially act as a CSE and any Bluetooth device with suitable IO-capabilities can act as a CCE.



*Figure 2.2: Example of CTN Profile Roles*

CTN objects may be received or sent by either the CSE or the CCE via an external network connection, (e.g., acellular network). The management of such external network connections is not covered by the CTN profile specification. The CTN Profile only requires that any CTN object received from, or sent to, an external network shall be stored in the object repository of the CSE.

## 2.3 User Scenarios

The following sections describe a set of technical use cases which may be implemented by CTN:

- **Select CTN account:**  
A user, or a CTN application on the CCE, selects a CTN account on the CSE. This is a basic functionality used to start a CTN session.
- **Download CTN listing:**  
The user wants to have overview information about the upcoming tasks, meetings, and notes.
- **Download selected CTN-entry:**  
A user, or a CTN client application, may need detailed information about a particular CTN-entry. For that, CTN-entries can be selectively downloaded.
- **Update notification:**  
The user wants to be notified about changes to the server's CTN repository (e.g., new meeting requests, meeting cancellations, meeting time change). The client device may register and de-register the notification service.
- **Reminder notification by server:**  
The user wants to be notified about calendar events or tasks (alarm notifications).

- When the CSE issues an alarm, a notification is sent to the CCE. The notification contains information required to inform the user about the upcoming event.
- An alarm notification from the CSE will initiate an alarm on the CCE and the user can react to it. Depending on the class of alarm, different types of reactions may be available (e.g., accept, deny, postpone).
- Optionally, the CCE may send a response to the CSE, which applies it to the internal handling of this alarm (update of the entry).
- Modify CTN- entry on CSE:  
The user, or the CTN client application, requests the CSE to update a CTN-entry, (e.g., update attributes or content).
- Delete CTN-entry on CSE:  
The user, or the CTN client application, requests the CSE to delete a CTN-entry.
- Upload a new CTN-entry:  
The user, or the CTN client application, requests the CSE to create a CTN-entry.
- Send CTN- entry:  
The user, or CCE wants to send, via the CSE, a calendar entry to a number of recipients.
- Confirm CTN-entry:  
The user or the CCE wants to confirm or reject a CTN-entry, (e.g., a meeting invitation). The confirmation can be sent to the recipient.
- Forward CTN-entry:  
The user, or the CCE, wants to forward a CTN-entry to additional recipients.

## 2.4 Safety Considerations

Since the use of CTN in a car is one of the most important usage scenarios, the implementer should consider safety in their user interface design, (e.g., to avoid driver distraction). In particular the CSE should switch off its acoustic alarm while connected with the CCE in a CTN session.

## 3 Application Layer

### 3.1 Overview

CTN defines the following object types that may be transferred between the CCE and CSE devices:

- **Literal objects (CTN-entry, bCalendar):**  
The actual PIM data objects stored in the repository of the CSE, calendar, task, and note entry objects. These objects are named CTN-entry objects in this specification while the term 'bCalendar' (see also Section 3.3.1) is used to name the CTN representation of a CTN-entry of any type, calendar, task, or note. Note that the real format of the object in the CSE's repository is out-of-scope for the CTN Profile.
- **Listing objects (CTN-listing):**  
Listings of literal objects with a number of listing entries, each with a limited but relevant amount of information about the related literal objects. These objects are named 'CTN-listing' objects in this specification.
- **Event report objects (CTN-event-report):**  
Events are sent by the CSE to the CCE to report changes in the CSE's object repository or changes of the status of literal objects.

For general requirements concerning the CTN application objects see GPP [6].

### 3.2 General Attributes and Conventions

#### 3.2.1 Character Set

The character set used for the attributes of the CTN-entry objects shall be UTF-8.

#### 3.2.2 Timestamps

All timestamps in CTN shall use local-time basis with UTC time offset and 24 hours representation. This applies both for timestamps on OBEX level and on the iCalendar objects transferred by CTN. For timestamps on OBEX level the format shall be "YYYYMMDDTHHMMSS±HH:MM" according to ISO 8601.

#### 3.2.3 Handles and Sessions

##### Handle values:

According to [6], the CCE and CSE shall use unique identifiers, called handles, to identify individual literal (bCalendar) objects. The CSE device shall assign a handle to each bCalendar object representing an CTN-entry object in its repository. The handle shall be a 128-bit unsigned integer whose value is defined by the CSE.

Each handle shall be locally unique within the context of the of a CAS account (see Section 3.5 for CAS account definition). This means that the handle, when presented to the CCE, shall unambiguously identify one and only one bCalendar object within in the related CAS account.

If a CCE is connected to two or more CAS accounts (on one or more CSEs), it is responsible for keeping track of the handles for each CAS account, as in such case-identical handles may exist, each in a different context.

#### Handle persistency:

Handles shall be persistent, (i.e., a handle value shall not be changed during the lifetime of a CTN-entry object on the CSE device). Thus, the CCE may retain and use a handle obtained in a previous CTN session.

A handle shall not be changed by the CSE, even if the related object is updated or modified. The CCE can verify that a locally stored CTN-entry object is current by checking the timestamp indicating the last update of the object.

See [6] Section 3 for further requirements on handle (e.g., Sessions hexadecimal representation).

#### Sessions:

A CTN session starts with the establishment of a CTN Access Service connection (see Section 4.1). The duration of the CTN session shall be defined as the time during which at least one CTN Access Service connection or CTN Notification Service connection is ongoing.

## 3.3 CTN Application Objects

### 3.3.1 bCalendar Objects (x-bt/Calendar)

Exchanged CTN-entry objects shall use the bCalendar format. The bCalendar object encapsulates the delivered CTN-entry objects that are encoded in iCalendar format [8] and provides additionally a suitable set of properties with helpful information. The general encoding characteristics as defined for iCalendar [8] and vCards [9] shall be applied.

The formal BNF definition of the bCalendar format is as follows:



```

<bcalendar-object> ::= {
    "BEGIN:BCAL"<CRLF>
        <version-property>
        <handle-property>
        <update-property>
        <bcal-content-property>
    "END:BCAL"<CRLF>
}

<version-property> ::= "VERSION:"
<common-digit>* "." <common-digit>* <CRLF>
<handle-property> ::= "HANDLE:" <hex-digit>* <CRLF>
<update-property> ::= "UPDATE:" <timestamp> <CRLF>
<bcal-content-property> ::= 'icalobject'
  
```

The following conventions shall be applied for the bCalendar properties:

1. version-property

The value for this property shall be "VERSION:1.0 ", which is the present bCalendar version 1.0 (see also [6]).

2. handle-property

"handle" is the bCalendar object handle in hexadecimal representation with 128-bit handles (see also Section 3.2.2 and [6]).

3. update-property

Timestamp of the last update on the CSE in the format specified in Section 3.2.2. For new objects, this timestamp is set to the creation time of the object. Any change of the related bCalendar object shall result in an update of the timestamp to the current CSE time.

4. bcal-content-property

The content-property is included in the iCalendar/vCalendar object according to [8]. The following requirements apply for CTN:

- The iCalendar object shall include exactly one of the following vCalendar components:
  - 'VEVENT' [8] in case of a calendar entry

- 'VTODO' [8] in case of a task entry
- 'VJOURNAL' [8] in case of a note entry
- All mandatory properties and parameters, according to [8], shall be supported both by the CSE and the CCE
- Optional properties and parameters may be delivered by the CSE and may be interpreted by the CCE. In any case, the CCE shall be able to accept and silently ignore properties and parameters that it does not implement.
- The delivered iCalendar object shall include at least the 'Attendee' [8] property representing the CSE with the participation status ('PARTSTAT' in [8]) to inform the CCE about the status and required actions of the related iCalendar object. The following values are allowed for CTN:
  - For calendar entries (VEVENT):
    - "NEEDS-ACTION": Event needs action (default)
    - "ACCEPTED": Event accepted
    - "DECLINED": Event declined
    - "TENTATIVE": Event tentatively accepted
    - "DELEGATED": Event delegated
  - For task entries (VTODO)
    - "NEEDS-ACTION": Task needs action (default)
    - "ACCEPTED": Task accepted
    - "DECLINED": Task declined
    - "TENTATIVE": Task tentatively accepted
    - "DELEGATED": Task delegated
    - "COMPLETED" ; Task completed
    - "IN-PROCESS" ; Task in process
  - For note entries (VJOURNAL)
    - "NEEDS-ACTION": Note needs action (default)
    - "ACCEPTED": Note accepted
    - "DECLINED": Note declined

A download of the bCalendar object shall not automatically change the status and the CCE has to actively request a status change (see setCTNStatus function in Section 5.5). Note that the status may be changed also by the CSE device during a CTN session (e.g., user changes it on the CSE device).

Below is an example of a bCalendar object of type VEVENT (calendar event):

```

BEGIN:BCAL
VERSION:1.0
HANDLE:00000000000000001234567890ABCDEF
UPDATE:20130701T163011+02:00
BEGIN:VCALENDAR
VERSION:2.0
BEGIN:VEVENT
UID:123@bms.com
ORGANIZER;CN="Joachim":MAILTO:joachim@generic.com
LOCATION:Chinese Tower
SUMMARY:Project summary
DESCRIPTION: Project meeting at the Chinese Tower garden
CLASS:PUBLIC
DTSTART;TZID=CET:20130707T183000
DTEND;TZID=CET:20130707T220000
DTSTAMP;TZID=CET:20130707T162900
ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=TENTATIVE;
CN=Kyle:mailto:kyle@gaudybird.com
END:VEVENT
END:VCALENDAR
END:BCAL"
  
```

### 3.3.2 CTN-Listing Object (x-bt/CTN-Listing)

The CTN-Listing object is an XML object and shall be encoded in UTF-8.

The entries of the CTN-Listing object shall be sorted by the value of the "datetime update" parameter (see below). Sort order shall be in descending order, with the newest object being transferred first.

In case of a recurrent series of events or tasks (e.g., weekly event) each single entry within the requested period shall be delivered while all properties of these events – except the datetime update parameter – shall have the same values (see also GetCTNListing function, Section 5.3).

The CTN-Listing object is defined according to the following DTD (Document Type Definition, [10]):

```
<!DTD for a CTN-Listing Object-->

<!DOCTYPE CTN-listing [

<!ELEMENT CTN-listing (ctn-entry)* >
<!ATTLIST CTN-listing version CDATA #FIXED "1.0">

<!ELEMENT ctn-entry (attachment)*>
<!ATTLIST ctn-entry
    handle CDATA #REQUIRED
    update CDATA #REQUIRED
    cal_type (event|task|note) #REQUIRED
    summary CDATA #IMPLIED
    starttime CDATA #REQUIRED
    endtime CDATA #IMPLIED
    originator_name CDATA #IMPLIED
    originator_address CDATA #IMPLIED
    size CDATA #REQUIRED
    priority (high|normal|low) "normal"
    pstatus CDATA #IMPLIED
    alarmstatus (on|no|off) "on"
    sendstatus (yes|no) "yes"
    recurrent (yes|no) "no"

>

<!ELEMENT attachment EMPTY>
<!ATTLIST attachment
    attach_id CDATA #REQUIRED
    attach_name CDATA #REQUIRED
    attach_type CDATA #REQUIRED
    attach_size CDATA #REQUIRED

>
]>
```

The following conventions shall be applied for the CTN-listing properties:

- "version" is the version of the CTN-listing object; the value shall be "1.0" for this specification; note that the CTN-listing version may be different from the version of the related specification.
- "handle" is the bCalendar object handle in hexadecimal representation (see Section 3.2.3).
- "update" is the timestamp of the last update in the format specified in Section 3.2.2. For new objects, this timestamp is set to the creation time of the object. Any change of the related bCalendar object shall result in an update of the timestamp to the current CSE time.
- "cal\_type" is the type of the bCalendar object; the value of this attribute shall be one of:
  1. "event" in case of an calendar event entry (type 'VEVENT' in [8]).
  2. "task" in case of a task entry (type 'VTOD0' in [8]).
  3. "note" in case of a note (entry (type 'VJOURNAL' in [8]).
- "summary" shall contain the summary or subject of the bCalendar object or, if not present, the first words of the description text. This attribute shall not exceed 256 bytes and shall be truncated if required.
- "starttime" is the start-timestamp of the bCalendar object in format specified in Section 3.2.2. The value of the attribute shall be:
  1. In case of an event, the start of the calendar event
  2. In case of a task, the start of the task
  3. In case of a note, the time of last update or change of the note
- "endtime" is the end-timestamp of the bCalendar object in format specified in Section 3.2.2. The value of the attribute shall be:
  1. In case of an event, the end of the calendar event.
  2. In case of a task, the due-time of the task.
  3. In case of a note, this property shall not be used.
- "originator\_name" is the name of the originator of the bCalendar object, if it is known by the CSE device and shall be omitted otherwise. This parameter shall not exceed 256 bytes and shall be truncated to the first 256 bytes if required.
- "originator\_address" is the email address of the originator of the bCalendar object, if it is known by the CSE device. This parameter shall not exceed 256 bytes and shall be truncated to the first 256 bytes if required.

- "size" is the size in bytes of the bCalendar object on the CSE side, including attachments, if present. The CCE implementation has to take into account that the size of the iCalendar entry delivered in a bCalendar object may be different from this size due to transcoding effects.
- "priority" is the priority of the bCalendar entry; allowed values are "high", "normal", or "low".
- "pstatus" indicates the status of the bCalendar entry. See description of the PARTSTAT parameter in Section 3.3.1 for values. The CCE may actively set this status (see setCTNStatus function in Section 5.5). Note that the status may also be changed on the CSE device during a CTN session.
- "alarmstatus" indicates whether the bCalendar entry has an active alarm or reminder. Possible values are:
  1. "on" means alarm is active.
  2. "no" means alarm is not active.
  3. "off" means alarm is de-activated.
  4. The CCE may actively set this status (see setCTNStatus function in Section 5.5). Note that the status may also be changed on the CSE device during a CTN session.
- "sendstatus" indicates whether the bCalendar entry has been sent to one or more of its participants. Value "yes" means it has been sent, value "no" means that the entry has not been sent to any participant. The CCE may actively set this status (see setCTNStatus function in Section 5.5). Note that the status may also be changed on the CSE device during a CTN session.
- "recurrent" indicates that the bCalendar entry belongs to a recurrent series of calendar events, tasks or notes, (e.g., a weekly event).
- The element "attachment" includes the attachments of the bCalendar object: One "attachment" element shall be present for each attachment where an attachment entry consists of the following attributes:
  1. "attach\_id" is an index of the attachment; the index shall have an unsigned integer value starting with '1' for the first attachment with an increment of one for each further attachment; the maximum value shall be 255.
  2. "attach\_name" is the name of the attachment if it is known by the CSE device. This parameter shall not exceed 256 bytes and shall be truncated to the first 256 bytes if required.
  3. "attach\_type" is the MIME type of the attachment according to [11].
  4. "attach\_size" is the size of the attachment in bytes on the CSE side. The CCE implementation has to take into account that the size of the attachment delivered in a bCalendar may be different from this size due to transcoding effects.

Below an example for a CTN-listing:

```
<CTN-listing version="1.0">
  <ctn-entry handle="123456789" update="20130510T173000+02:00"
cal_type="event" summary="project meeting Aumeister"
starttime="20130510T193000+02:00" endtime="20130510T220000+02:00"
originator_name="Joachim" originator_address="joachim@generic.com"
size="450" priority="high" pstatus="needs-action" alarmstatus="on"
sendstatus="yes" recurrent="no" />
  <ctn-entry handle="000000000000000000000000000000002345"
update="20130408T123222+02:00" cal_type="event" summary="project
summary Havanna Bar" starttime="20130510T220000+02:00"
endtime="20130511T020000+02:00" originator_name="Dominik"
originator_address="dominik@generic.com" size="395" priority="high"
pstatus="accepted" alarmstatus="off" sendstatus="yes" recurrent="no"
/>
</CTN-listing>
```

### 3.3.3 CTN-Event-Report Object (x-bt/CTN-EventReport)

The CTN-event-report object is an XML object including the description of exactly one event. The CTN-event-report object shall be encoded in UTF-8.

The CTN-event-report object is defined according to the following DTD:



```

<!DTD for the CTN-EventReport object-->

<!DOCTYPE CTN-event-report [

<!ELEMENT CTN-event-report ( event ) >
<!ATTLIST CTN-event-report version CDATA #FIXED "1.0">

<!ELEMENT event EMPTY>
<!ATTLIST event
    type CDATA #REQUIRED
    handle CDATA #REQUIRED
    cal_type CDATA #REQUIRED
    summary CDATA #REQUIRED
    update CDATA #REQUIRED
    originator_name CDATA #IMPLIED
    originator_address CDATA #IMPLIED
>
]>

```

The following conventions shall be applied for the CTN-event-report properties:

- "version" is the version of the CTN event-report object; the value shall be "1.0" for this specification; note that the version may be different from the version of the related specification.
- "type" is the type of the event, indicating the related change in the CSE repository:
  1. "NewObject": indicates that a new CTN-entry object has been added to a folder of the CSE device.
  2. "ObjectUpdate": indicates that an existing object has been modified on the CSE device. The update may include any change of properties or states of the object, (e.g., new/removed recipients, confirmation by recipients). This event type shall also be used if a new object completely replaces an old one.
  3. "Alarm" indicates that an alarm, or reminder of a bCalendar, entry has been triggered at the CSE, (e.g., the start time of a calendar or task entry).
  4. "SendingSuccess": indicates that a bCalendar entry has been successfully sent to a remote network. Only applicable if the related bCalendar entry has to be sent to other participants.
  5. "SendingFailure": indicates that sending to a remote network failed.

6. "DeliverySuccess": indicates that a bCalendar object has been successfully delivered to its intended recipient.
  7. "DeliveryFailure": indicates that the delivery of a bCalendar object to its intended recipient failed.
  8. "MemoryFull": indicates that the memory of the CSE device is full. The CSE will not be able to add or receive new bCalendar objects until this situation is resolved.
  9. "MemoryAvailable": indicates that the memory of the CSE device is available (again) so new bCalendar objects can be added or received. The event shall be sent when the condition that caused "MemoryFull" event has been resolved. Additionally, the event may be sent at the beginning of a CTN session.
  10. "ObjectDeleted": indicates that a bCalendar object of the reported type has been deleted on the CSE.
- "handle" is the related bCalendar object handle in hexadecimal representation (see Section 3.2.3)
  - "cal\_type" is the type of the bCalendar object (see "type" attribute in CTN-listing object, Section 3.3.2)
  - "summary " is the summary or subject of the bCalendar object (see "summary" attribute in CTN-listing object, Section 3.3.2)
  - "update " is the timestamp of the last update of the bCalendar object (see also "update" attribute in CTN-listing object, Section 3.3.2)
  - "originator\_name" and "originator\_address" is the originator name and email address of the bCalendar object (see "originator\_name" and "originator\_address" attribute in CTN-listing object, Section 3.3.2). The originator equals to the 'CHAIR' participant role in the iCalendar definition [8].

The properties above shall be used in the following way for the particular notification-event types:

type	version	handle	cal_type	summary	update	originator_name	originator_address
NewObject	M	M	M	M	M	M	M
ObjectUpdate	M	M	M	M	M	M	M
Alarm	M	M	M	M	M	O	O
SendingSuccess	M	M	O	O	O	O	O
SendingFailure	M	M	O	O	O	O	O
DeliverySuccess	M	M	O	O	O	O	O
DeliveryFailure	M	M	O	O	O	O	O
MemoryFull	M	X	X	X	X	X	X
MemoryAvailable	M	X	X	X	X	X	X
ObjectDeleted	M	M	M	O	O	O	O

**Table 3.1:** Usage of properties in CTN notification event

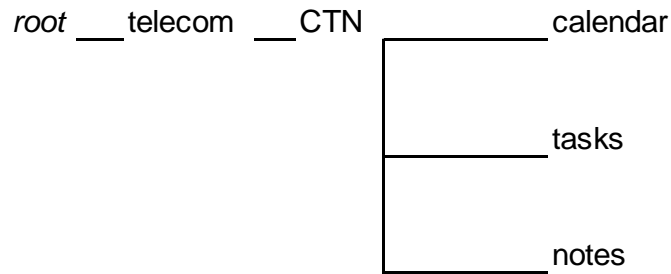
Below an example for a CTN event-report:

```
<CTN-event-report version="1.0">
<event type="NewObject" handle="12345678" cal_type="event"
summary="we have to talk" update="20130501T200000+01:00"
originator_name="Ruediger" originator_address="ruediger@generic.com"
/>
</CTN-event-report>
```

### 3.4 CTN Folder-Structure

CTN uses a fixed folder structure following the general conventions for folders as defined in [6].

The following figure shows the structure of the CTN repository as presented by the CSE. Note that the CTN folder structure below is a virtual structure that may differ from the physical repository of the specific CSE device. Each CTN CAS account shall present its own virtual folder structure (see also Section 3.5)



**Figure 3.1:** CTN virtual folder structure

Description:

- **telecom/CTN/calendar:**  
If the CSE supports calendar functions this folder shall contain all calendar entries related to the corresponding account.
- **telecom/CTN/tasks:**  
If the CSE supports tasks functions this folder shall contain all tasks entries related to the corresponding account.
- **telecom/CTN/notes:**  
If the CSE supports notes functions this folder shall contain all notes entries related to the corresponding account.

### 3.5 CTN Access Server (CAS) Accounts

A CSE device may present one or several instances – called CAS accounts hereafter – to the CCE, each providing the overall CSE server functionality. The CAS accounts follow the conventions stated in GPP [6] for CSE instances (Section 3.3.3).

For each CAS account there shall be exactly one CAS-Server SDP record (see Section 7.2). Each CSE account shall have an ID unique for the CTN profile. A CCE shall access to each CAS account by a dedicated OBEX connection.

## 4 CTN Profile Features

### 4.1 Overview

As described in Section 2.1 CTN is based on OBEX [3]. For further information about the OBEX/GOEP requirements see GPP [6]. Within the scope of CTN, the following OBEX services are defined:

- The CTN Access Service (CAS) is an OBEX service by which the CCE acts as an OBEX Client and connects to a CSE that acts as an OBEX Server.
- The CTN Notification Service (CNS) is an OBEX service by which the CSE acts as an OBEX Client and connects to the CCE that acts as an OBEX Server.

CTN includes the features in the table below. For a device to comply with this specification, it shall observe the following implementation requirements table:

Feature	Support by the CCE	Support by the CSE
CTN Account Management	M	M
CTN Notification	C1	M
CTN Browsing	C1	M
CTN Downloading	M	M
CTN Uploading	O	O
CTN Forward	O	O
CTN Delete	O	O

**Table 4.1:** CTN features

C1: At least one of these features shall be supported by device

### 4.2 CTN Account Management Feature

If support for the CTN Account Management feature is claimed by the CCE this feature allows the CCE to manage the account on the CSE by requesting account information or initiating account synchronization:

Function	Support by the CCE	Support by the CSE
GetCTNAccountInformation	M	M
CTNNotificationRegistration	C1	M
SyncCTNAccount	O	C2

C1: Mandatory if CCE supports CTN Notification feature

C2: Mandatory if CSE CTN account supports triggering of synchronization with an external server

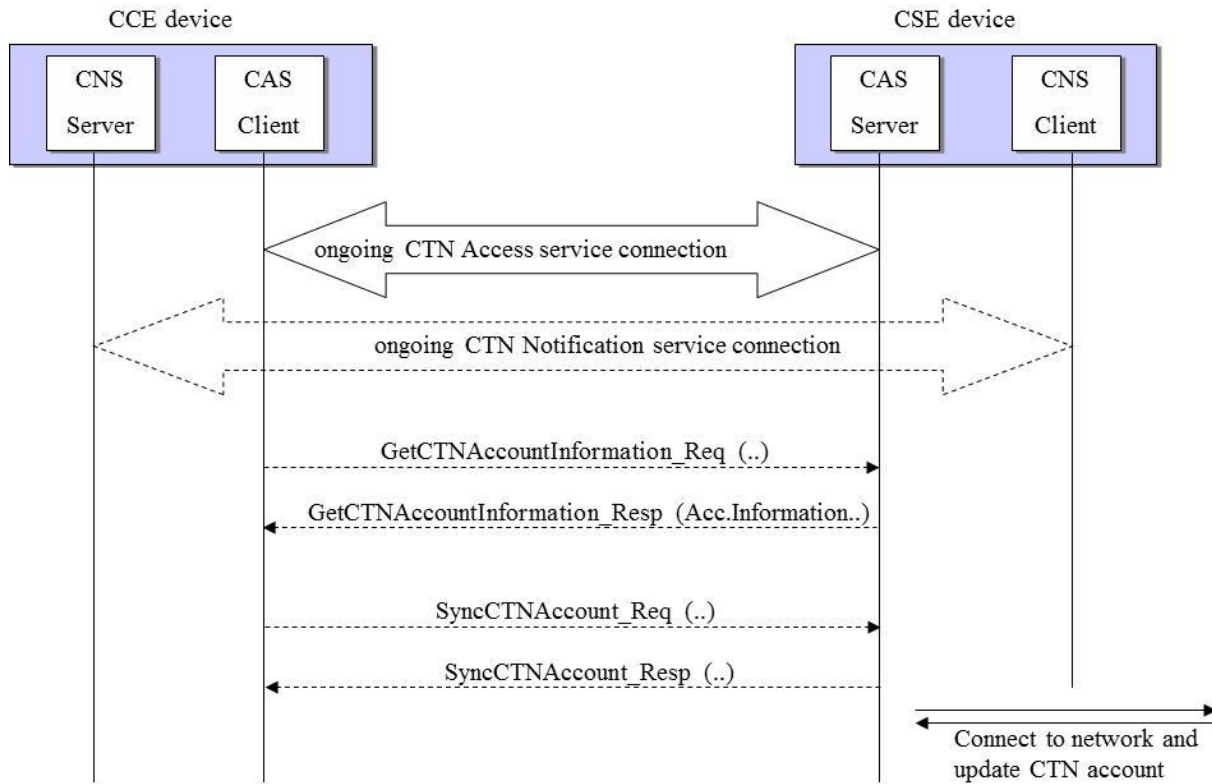


Figure 4.1: Example sequence for the functions of the CTN Account Management feature

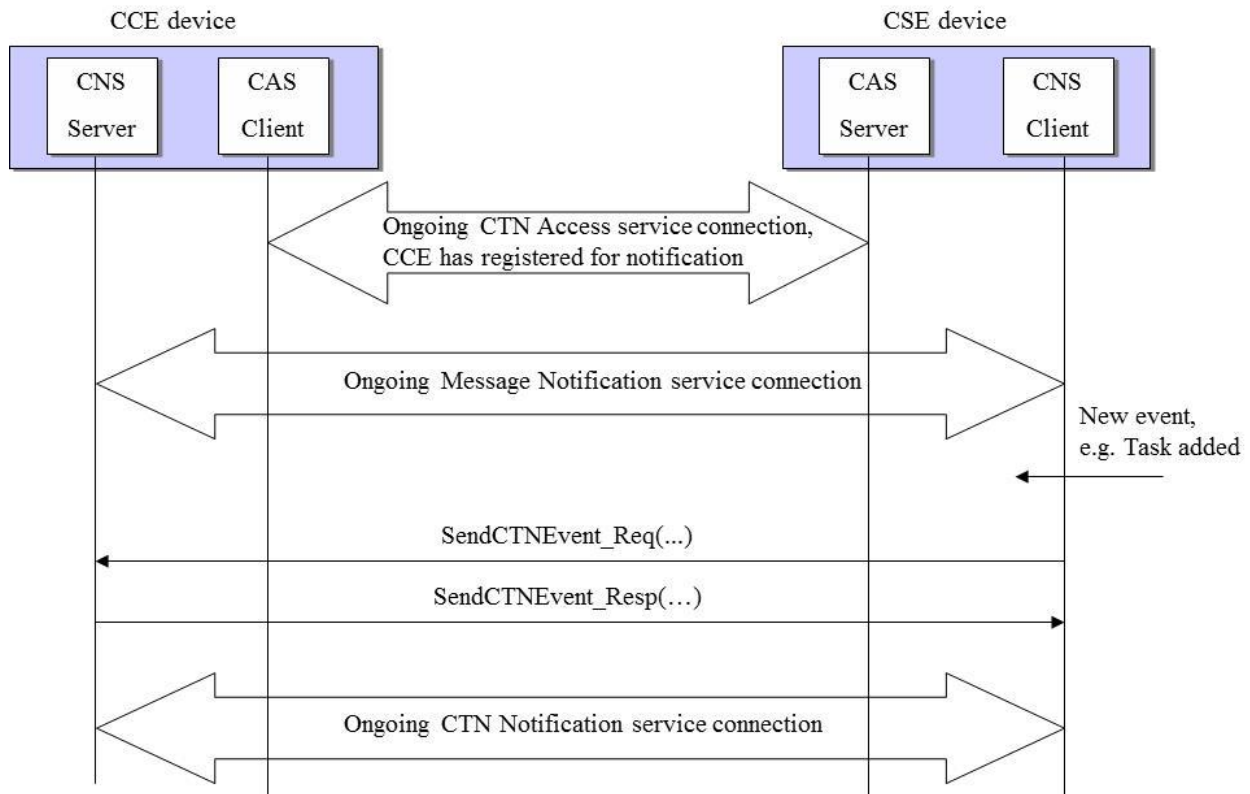
### 4.3 CTN Notification Feature

If support for the CTN Notification feature is claimed by the CCE, this feature allows the CSE to send event reports to the CCE, (e.g., arrival of new entries, changes of entries or alarm events):

Function	Support by the CCE	Support by the CSE
SendCTNEvent	M	M

The CSE shall notify the CCE about any change of within its repository that was not initiated by the CCE, by a dedicated event.

The mode of the notification may be configured by the CCE using the `CTNNotificationRegistration` function (See Section 5.2). The default mode for the notification is "off", (i.e., the notification is inactive at the CSE when the CTN session starts).



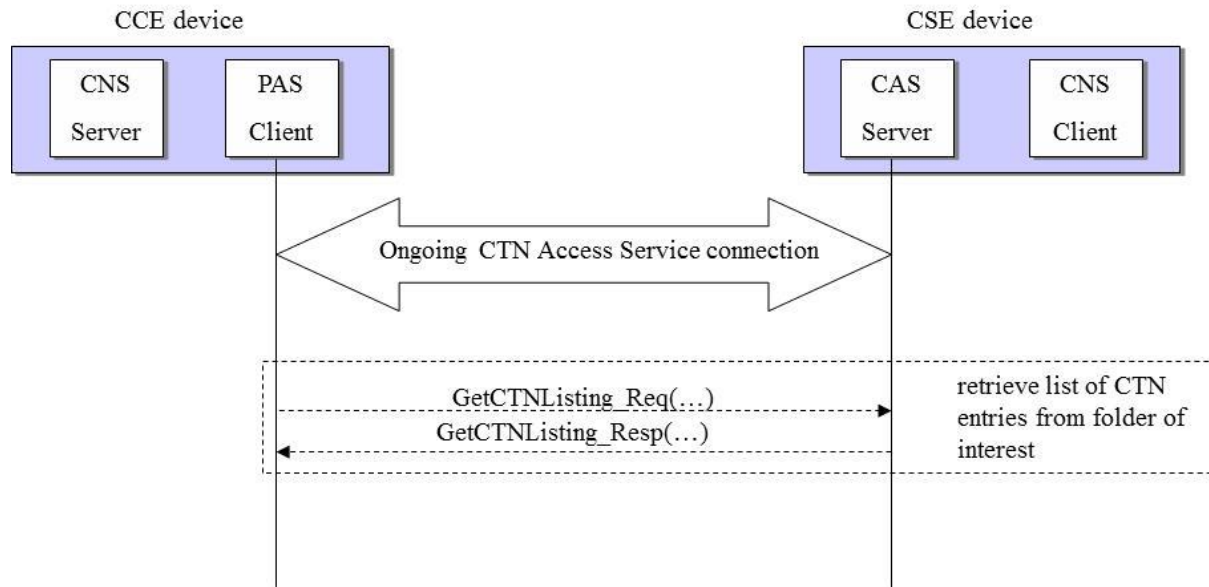
**Figure 4.2:** Sequence for the CTN Notification feature

## 4.4 CTN Browsing Feature

This feature allows the CCE to browse on the CSE:

Function	Support by the CCE	Support by the CSE
GetCTNListing	M	M





**Figure 4.3:** Sequence for the CTN Browsing feature

## 4.5 CTN Downloading Feature

This feature allows the CCE to retrieve a bCalendar object from the CSE:

Function	Support by the CCE	Support by the CSE
GetCTNObject	M	M
SetCTNStatus	M	M

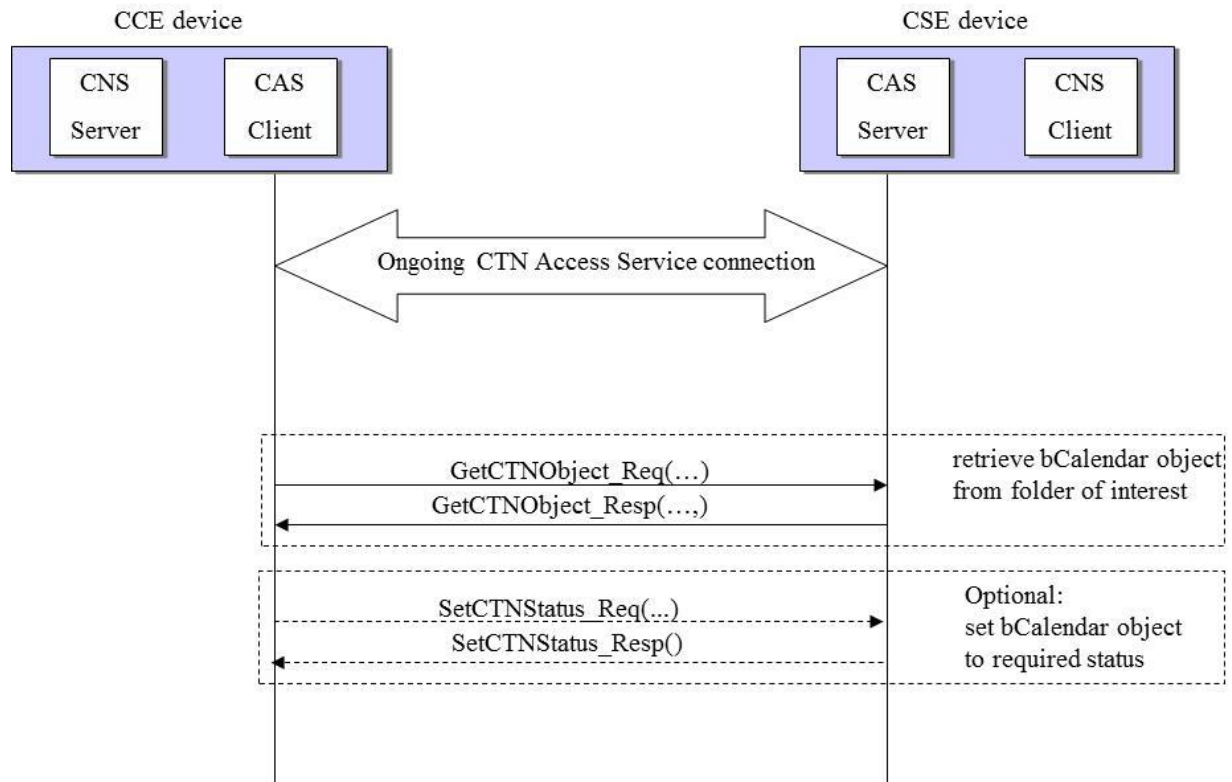


Figure 4.4: Sequence for the CTN Downloading feature

## 4.6 CTN Uploading Feature

This feature allows the CCE to upload a bCalendar object to the CSE. Dependent on the parameters of the `PushCTNObject` function this may also include sending of a CTN-entry object to an external network:

Function	Support by the CCE	Support by the CSE
<code>PushCTNObject</code>	M	M

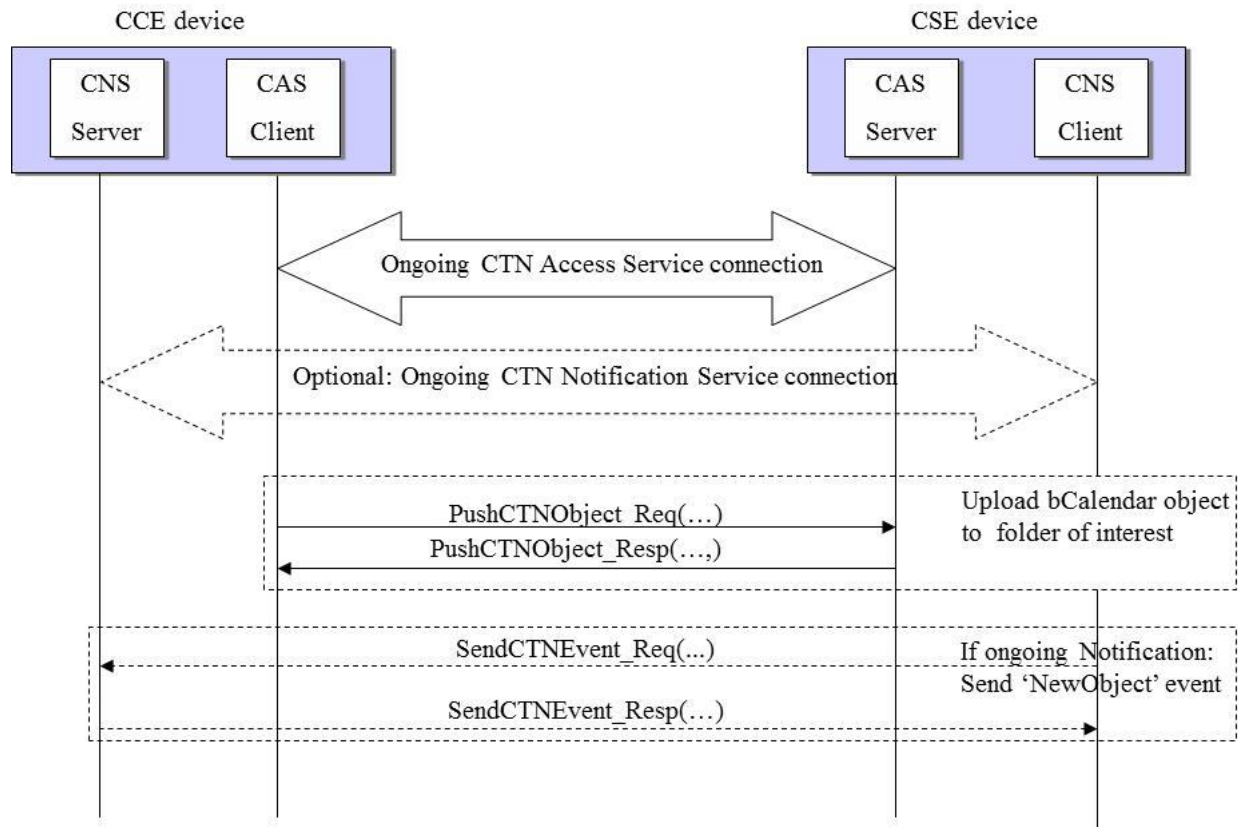


Figure 4.5: Sequence for the CTN Uploading feature

## 4.7 CTN Forward Feature

This feature allows the CCE to forward a bCalendar entrie that resides in the CSE's repository:

Function	Support by the CCE	Support by the CSE
ForwardCTNObject	M	M

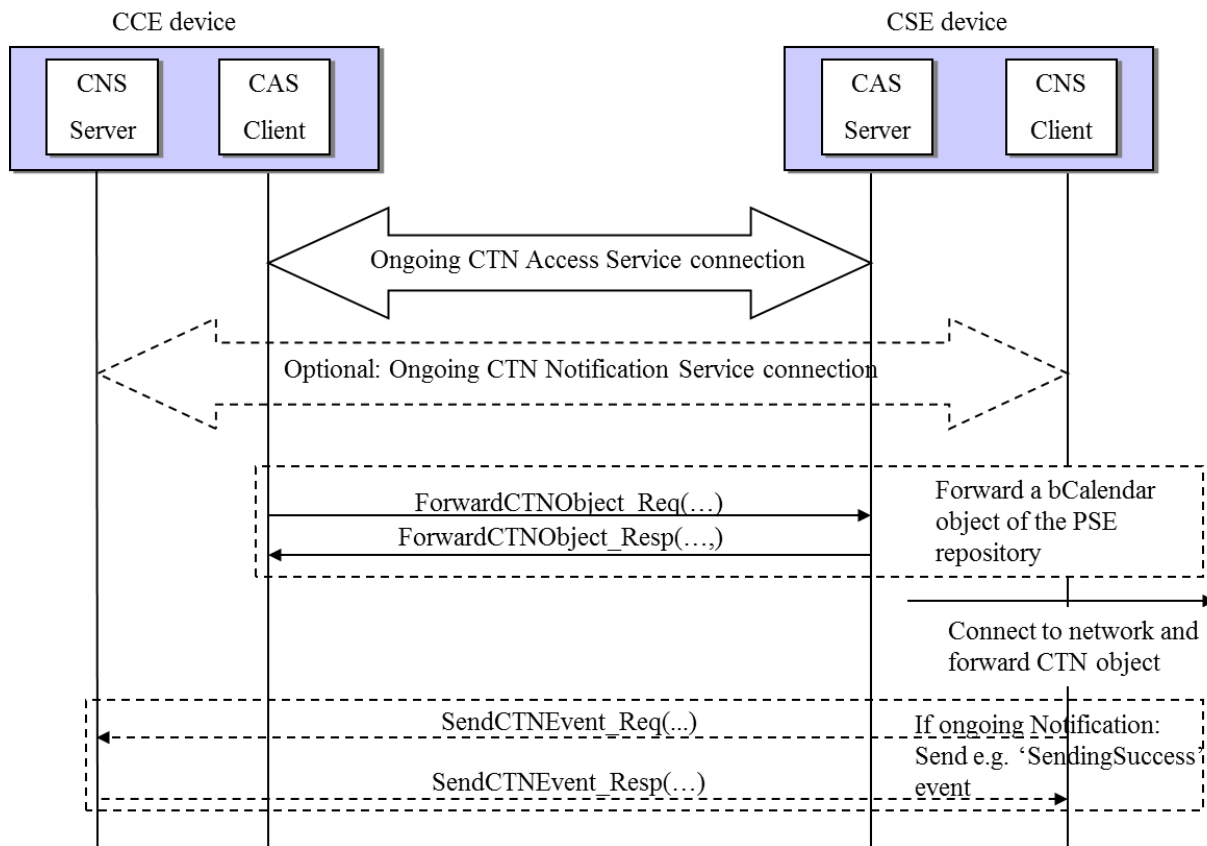


Figure 4.6: Sequence for the CTN Forward feature

## 4.8 CTN Delete Feature

This feature allows the CCE to delete a bCalendar entry that resides in the CSE's repository:

Function	Support by the CCE	Support by the CSE
SetCTNStatus	M	M

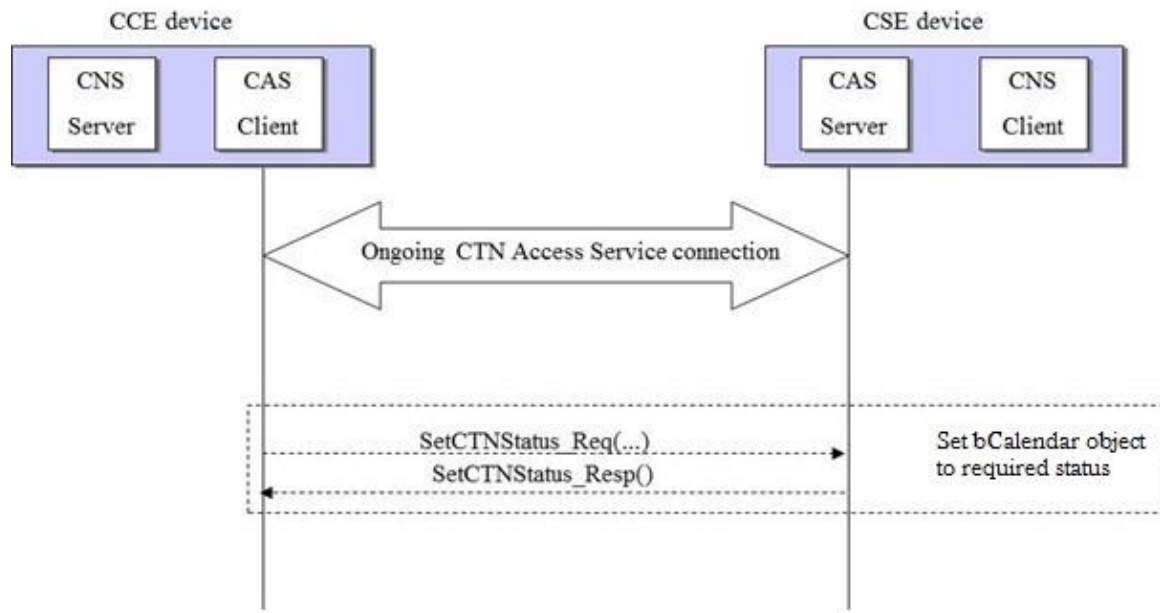


Figure 4.7: Sequence for the CTN Delete feature

## 4.9 Initialization sequences

The OBEX connection initialization mechanisms are described in GPP ([6] Section 4.2) where the CTN Access Service shall follow the conventions of the GPP PIM Access Service and the CTN Notification Service shall follow the conventions of the GPP PIM Notification Service.

## 5 CTN Profile Functions

### 5.1 SendCTNEvent function

This function is used by the CSE to send a notification event to the CCE when an alarm triggers or when there is a change in the CSE's object repository, (e.g., an addition, a removal, a modification or a status change of a CTN-entry object in the repository). The function follows the general definitions of the GPP SendEvent function [6].

Each change within the repository shall result in a dedicated event-sending-action by the CSE device.

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	PUT (0x02 or 0x82)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Single Response Mode	0x01	M
Header	Type	x-bt/CTN-EventReport	M
Header	Application Parameters		
	- InstanceID	Varies	M
Header	Body/End of Body	CTN-EventReport Object	M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M
Field	Packet Length	Varies	M
Header	Single Response Mode	0x01	C1
Header	Single Response Mode Param	0x01	C2

C1: The Single Response Mode header is mandatory (M) if:

- an SRM header has been received in the previous PUT request AND
  - the response code is success (0x90 or 0xA0).
- otherwise excluded (X).

C2: The Single Response Mode Param header is optional (O) if Single Response Mode is used, otherwise excluded (X)

#### 5.1.1 Connection ID

The connection ID header shall be used to indicate the connection ID according to [6].

### 5.1.2 Type

The type header shall be used to indicate the type of object to be transmitted:

<x-bt/CTN-EventReport> CTN-Event-Report object

### 5.1.3 Application parameters

For further details about the application parameters, see Section 6.2.4.

#### InstanceID

This header shall be used by the CSE to indicate the corresponding 'CTNAccountID' of the CTN Account (see Sections 3.5 and 7.2). As only one CNS connection can be established from the CSE device to the CCE this parameter is required by the CCE to identify the CAS Account that generated this event. For further details see Section 6.2.4.

### 5.1.4 Body/EndOfBody

These headers shall contain the CTN-event-report object that is sent by the CSE device, see Section 3.3.3.

## 5.2 CTNNotificationRegistration function

This function is used by the CCE device to register for notifications indicating changes in the CSE's object repository. The function follows the general definitions of the GPP SetNotificationRegistration function [6].

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	PUT (0x02 or 0x82)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Type	x-bt/CTN-NotificationRegistration	M
Header	Application Parameters		
	- NotificationStatus	on/off	M
	- AcousticAlarmStatus	on/off	M
Header	Body/End of Body	Filler-byte 0x30	M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M
Field	Packet Length	Varies	M



### 5.2.1 Connection ID

The Connection ID header shall be used to indicate the connection ID according to [6].

### 5.2.2 Type

The type header shall be used to indicate the type of object to be transmitted:

<x-bt/CTN-NotificationRegistration>

### 5.2.3 Application parameters

For further details about the application parameters, see Section 6.2.4.

#### NotificationStatus

The CCE shall indicate the request for being notified about changes in the object repository. According to [6] the header shall have either of the values

- "off", meaning no notification required
- "on", meaning the CTN Notification Service (CNS) shall be established

Default value on CSE side after start of the CTN session shall be "off"

#### AcousticAlarm

The CCE shall use this header to switch the acoustic alarm state for CTN-Entries on the CSE side

- "off", meaning suppress the acoustic alarm on the CSE device
- "on", meaning an acoustic alarm is allowed (but not mandatory) on the CSE device

Default value on CSE side after start of the CTN session shall be "on"

### 5.2.4 Body/EndOfBody

To avoid PUT with empty Body leading to a 'delete' of the related CTN entry, these headers shall contain a filler byte. The value of this byte shall be 0x30 (= "0").

## 5.3 GetCTNListing function

This function is used by the CCE to retrieve the listing of bCalendar objects by the CSE. The function follows the general definitions of the GPP GetObjectListing function [6].

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	GET (0x03 or 0x83)	M
Field	Packet Length	Varies	M

Header	Connection ID	Varies	M
Header	Single Response Mode	0x01	M
Header	Single Response Mode Param	0x01	O
Header	Type	x-bt/CTN-Listing	M
Header	Name	Name of the Folder	M
Header	Application Parameters		
	- MaxListCount	Varies	O
	- ListStartOffset	Varies	O
	- FilterPeriodBegin	Varies	O
	- FilterPeriodEnd	Varies	O
	- ParameterMask	Varies	O

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M
Field	Packet Length	Varies	M
Header	Single Response Mode	0x01	C1
Header	Single Response Mode Param	0x01	C2
Header	Application Parameters		
	- ListingSize	Varies	C3
	- CSETime	Varies	C3
Header	Body/End of Body	CTN-Listing object	C3

- C1: The Single Response Mode header is mandatory (M) if:
- an SRM header has been received in the previous GET request AND
  - the response code is success (0x90 or 0xA0).
- otherwise excluded (X).
- C2: The Single Response Mode Param header is optional if Single Response Mode is used, otherwise excluded (X).
- C3: These parameters shall be present if the request has been successful.

### 5.3.1 Connection ID

The ConnectionConnection ID header shall be used to indicate the connection ID according to [6].

### 5.3.2 Type

The type header shall be used to indicate the type of object to be transmitted:

<x-bt/CTN-Listing>

### 5.3.3 Name

This property shall be used to indicate the folder from which the CTN-listing object is to be retrieved. The value of this property shall be a folder path according to the Description in Section 3.4.

### 5.3.4 Application parameters

For further details about the application parameters, see Section 6.2.4.

#### MaxListCount

This application parameter header may be used to indicate the maximum number of bCalendar entries listed in the CTN-listing object. For further information see definition of GetObjectListing function in [6].

#### ListStartOffset:

This application parameter header may be used to indicate the offset of the first entry of the returned CTN-listing object. For further information see definition of GetObjectListing function in [6].

#### FilterPeriodBegin, FilterPeriodEnd:

These application parameters headers may be used by the CCE to filter the entries returned in the CTN-listing object by the “datetime”. The following requirements apply:

- The two parameters shall be formatted according to Section 3.2.2.
- If one or both of the parameters are not defined by the user, the client device shall use an implementation defined default value for the filter period begin and/or end to avoid an infinite number of listing entries, (e.g., in case of a recurring event).
- If the value of "FilterPeriodBegin" is larger than the value of "FilterPeriodEnd" no entries shall be delivered.
- The ‘datetime’ attribute in the CTN-listing shall be the basis for the filtering.

#### **Annotation Filtering** (attributes above):

If list-segmentation with MaxListCount/ListStartOffset is requested in combination with filter-operations, then the filter-operations shall be applied first. The segmentation shall subsequently be applied onto the filtering result, which shall be a list, sorted chronologically in descending order of the ‘datetime’ attribute in the CTN-listing.

#### AttributeSelector

This application parameter header may be used by the CCE to determine the optional parameters contained in the requested CTN-Listing object.

Attributes and elements of the CTN-Listing object DTD (Section 3.3.2) labeled as "REQUIRED" shall be always present in the CTN-Listing. Bit  $i=1$  of the mask below indicates that the parameter related to Bit  $i$  shall be present, Bit  $i=0$  indicates that the parameter related to Bit  $i$  shall not be present in the requested CTN-Listing. The reserved bits shall be set to 0 by CCE, and ignored by CSE.

If this header is not specified the CSE shall return all parameters of the CTN-Listing object DTD labeled as "REQUIRED", and may return other attributes (implementation defined).

bit 0	attachment
bit 1	summary
bit 2	endtime
bit 3	originator_name
bit 4	originator_address
bit 5	priority
bit 6	pstatus
bit 7	alarmstatus
bit 8	sendstatus
bit 9	recurrent
bit 10-31	Reserved for future use

### ListingSize

If the request has been successful, this application parameter shall be included in the response to report the number of bCalendar objects that are selected by the filter parameters described above. For further information see definition of GetObjectListing function in [6].

### CSETime:

If the request has been successful, this application parameter shall be included in the response to report the Local Time basis of the CSE and its UTC offset. This information allows the CCE to interpret the timestamps of the CTN listing entries. The format shall be according to Section 3.2.2, (e.g., for Central European Time (CET) the offset is UTC+1h so the delivered value is "YYYYMMDDTHHMMSS+01:00" where "YYYYMMDDTHHMMSS" is the current CET timestamp of the CSE). If the CSE device has no information about the current UTC time, the offset value shall not be included so the delivered parameter is "YYYYMMDDTHHMMSS".

### **5.3.5 Body/EndOfBody**

If the request has been successful, these headers shall contain the CTN-listing object that is returned by the CSE device. For further information see definition of GetObjectListing function in [6].

## 5.4 GetCTNObject function

This function is used by the CCE to download a bCalendar object from the CSE's repository. The function follows the general definitions of the GPP GetObject function [6].

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	GET (0x03 or 0x83)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Single Response Mode	0x01	M
Header	Single Response Mode Param	0x01	O
Header	Name	bCalendar object handle	M
Header	Type	x-bt/Calendar	M
Header	Application Parameters		
	- Recurrent	yes/no	O
	- Attachment	on/off /select	M
	- AttachmentID	Varies	C1

C1: Mandatory (M) if Value of 'Attachment' attribute is 'select', excluded otherwise

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M
Field	Packet Length	Varies	M
Header	Single Response Mode	0x01	C1
Header	Single Response Mode Param	0x01	C2
Header	Body/End of Body	bCalendar object	C3

C1: The Single Response Mode header is mandatory (M) if:

- an SRM header has been received in the previous GET request AND
  - the response code is success (0x90 or 0xA0).
- otherwise excluded (X).

C2: The Single Response Mode Param header is optional if Single Response Mode is used, otherwise excluded (X).

C3: These parameters shall be present if the request has been successful

### 5.4.1 Connection ID

The Connection ID header shall be used to indicate the connection ID according to [6].

### 5.4.2 Type

The type header shall be used to indicate the type of object to be transmitted:

<x-bt/Calendar>

### 5.4.3 Name

The Name header shall be used to indicate the handle of the bCalendar object to be retrieved. For further information see definition of GetObject function in [6].

### 5.4.4 Application parameters

For further details about the application parameters, see Section 6.2.4.

#### Recurrent

This application parameter shall be used in case recurrent events or task to indicate whether a single event or the overall series shall be delivered. The parameter value shall be either:

- "yes" overall series to be delivered in the requested iCalendar object
- "no" only single event to be delivered

#### Attachment

This application parameter shall be used if the attachment(s) (if any) of a bCalendar object are to be included in the bCalendar object returned by the CSE. Allowed values of this parameter are:

- "off" - no attachments are delivered
- "on" - all attachments are delivered
- "select" - a selected attachment is requested, see 'Attachment ID' attribute below

#### AttachmentID

This application parameter shall be present if the value 'Attachment' Attribute is "select". It provides the ID of the attachment that shall be included in the requested bCalendar object ('attach\_id' of the CTN-Listing object, see Section 3.3.2). Note that only one attachment can be selected by this mechanism.

### 5.4.5 Body/EndOfBody

If the request has been successful, these headers shall contain the body of the bCalendar object that is returned by the CSE device. The object may be in its original form, or it may have been stripped of all its attachments, depending on the value of the application parameter "Attachment" in the request.

## 5.5 SetCTNStatus function

The function allows the CCE to modify the status of a bCalendar entry on the CSE.

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	PUT (0x02 or 0x82)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Type	x-bt/CalendarStatus	M
Header	Name	Handle of bCalendar object	M
Header	Application Parameters		
	- StatusIndicator	pStatus/alarmStatus/sendStatus/deletedStatus	M
	- StatusValue	Varies	M
	- PostponeVal	Varies	C1
Header	Body/End of Body	Filler-byte 0x30	M

C1: Mandatory (M) if StatusIndicator is "alarmStatus" and StatusValue is "postpone"; otherwise excluded (X)

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M
Field	Packet Length	Varies	M

### 5.5.1 Connection ID

The Connection ID header shall be used to indicate the connection ID according to [6].

### 5.5.2 Name

The Name header shall contain the handle of the bCalendar event whose status shall be modified. The handle shall be represented by a null-terminated Unicode text string with up to 32 hexadecimal digits.

### 5.5.3 Type

The type header shall be used to indicate the type of object to be sent.

<x-bt/CalendarStatus>

### 5.5.4 Application Parameters

#### StatusIndicator:

This application parameter header shall be used to indicate which status information is to be modified. The parameter value shall be either

- "pStatus" for the participant status of the participant related to the CSE or

- "alarmStatus" for the alarm status
- "sendStatus" for the send status
- "deletedStatus" for the delete status

#### StatusValue:

This header shall be used to indicate the new value of the status indicator to be modified. The parameter shall have one of these values:

- "needs-action", "accepted", "declined", "tentative", "delegated", "completed", "in-progress" for the "pStatus" indicator (see 'PARTSTAT' parameter in Section 3.3.1 for conventions for the particular bCalendar types)
- "yes" (=activate alarm), "no" (=deactivate) or "postpone" (=postpone) for the "alarmStatus" indicator
- "yes" (=send to participants), "no" (=don't send to participants) for the "sendStatus" indicator
- "yes" (deleted) for the "deletedStatus" indicator

Changing the status of a bCalendar entry can initiate actions on the CSE:

- bCalendar entries whose alarmStatus have been set to "yes" (=activated) shall send an alarm notification according to the conditions of the related bCalendar event (e.g., due datetime not in the past).
- bCalendar entries whose alarmStatus have been set to "no" (=deactivated) shall not send, or discontinue sending, alarm notifications.
- bCalendar entries whose alarmStatus have been set to "postpone" shall transmit another alarm notification according to the value of the PostponeVal application parameter header (see below).
- bCalendar entries whose sendStatus have been set to "yes" (=send) shall send the related entry to the related participants. The implementation of this transmission is out-of-scope for CTN. If the transmission to at least one of the participants is not possible for any reason, the CSE shall send a "SendingFailure" notification event to the CCE.
- bCalendar entries whose deletedStatus have been set to "yes" (=delete) shall be deleted.
- For bCalendar entries whose confirmStatus has been changed the CSE device shall send a confirmation to the originator of the entry. The implementation of this confirmation is out-of-scope for CTN. If the confirmation transmission is not possible for any reason the CSE shall send a "SendingFailure" notification event to the CCE.



### PostponeVal:

This application parameter header shall be used to indicate the number of minutes the alarm shall be postponed. The value shall be a positive integer.

### 5.5.5 Body/EndOfBody

To avoid PUT with empty Body leading to a 'delete' of the related bCalendar entry, these headers shall contain a filler byte. The value of this byte shall be 0x30 (= "0").

## 5.6 PushCTNObject function

This function is used by the CCE to upload a bCalendar object to the CSE's repository. The function follows the general definitions of the GPP PushObject function [6].

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	PUT (0x02 or 0x82)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Single Response Mode	0x01	M
Header	Type	x-bt/Calendar	M
Header	Name	Name of the Folder	M
Header	Application Parameters		
	- Send	yes/no	M
Header	Body/End of Body	bCalendar object	M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M
Field	Packet Length	Varies	M
Header	Single Response Mode	0x01	C1
Header	Single Response Mode Param	0x01	C2
Header	Name	Handle of the Object	C3

C1: The Single Response Mode header is mandatory (M) if:

- an SRM header has been received in the previous PUT request AND
  - the response code is success (0x90 or 0xA0).
- otherwise excluded (X).

C2: The Single Response Mode Param header is optional if Single Response Mode is used, otherwise excluded (X).

C3: These parameters shall be present if the request has been successful.

### 5.6.1 Connection ID

The Connection ID header shall be used to indicate the connection ID according to [6].

### 5.6.2 Name

In a request, this property shall indicate the folder to which the bCalendar object is to be pushed or, in case of a response, the handle of the bCalendar entry assigned by the CSE. See PushObject function in [6] for further information.

### 5.6.3 Type

The type header shall be used to indicate the type of object to be sent.

<x-bt/Calendar>

### 5.6.4 Application parameters

#### StatusIndicator:

This header shall be used to indicate whether a bCalendar uploaded to the CSE shall be sent to its participants. The parameter value shall be either

- "yes" - store and send to participants
- "no" - store but don't send to participants

If the transmission to at least one of the participants is not possible for any reason, the CSE shall send a "SendingFailure" notification event to the CCE.

### 5.6.5 Body/EndOfBody

If the request has been successful, this header shall contain the body bCalendar object that is pushed to the CSE device, see Section 3.3.1.

## 5.7 ForwardCTNObject function

This function is used by the CCE to forward a bCalendar of the CSE's repository to one or more additional recipients.

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	PUT (0x02 or 0x82)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Type	x-bt/CTN-forward	M
Header	Name	Handle	M

Header	Description	Varies	M
Header	Body/End of Body	Filler-byte 0x30	M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M
Field	Packet Length	Varies	M

### 5.7.1 Connection ID

The Connection ID header shall be used to indicate the connection ID according to [6].

### 5.7.2 Name

The Name header shall include the handle of the bCalendar object to be forwarded.

### 5.7.3 Type

The type header shall be used to indicate the type of the operation performed by the CSE.

<x-bt/CTN-forward>

### 5.7.4 Description

This header shall contain the list of recipients for the forwarded bCalendar object. The recipient list is a Unicode text string and shall include one or more SMTP email-addresses. When multiple addresses are specified, they are separated by semicolons.

The object related to the bCalendar object shall be forwarded by the CSE to the added recipient(s) if possible. Note that this may be not always possible, (e.g., due to network problems). The CSE shall transmit a "SendingSuccess" notification event to the CCE if the send to the external network has been successful. If the transmission to at least one of the participants is not possible for any reason, the CSE shall send a "SendingFailure" notification event to the CCE.

### 5.7.5 Body/EndOfBody

To avoid PUT with empty Body leading to a 'delete' of the related bCalendar entry, these headers shall contain a filler byte. The value of this byte shall be 0x30 (= "0").

## 5.8 GetCTNAccountInformation function

This function may be used by the CCE to retrieve user-readable information about the CTN accounts (or instances) provided by the CSE. The function follows the general definitions of the GPP GetInstanceNameObject function [6].

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	GET (0x03 or 0x83)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Single Response Mode	0x01	M
Header	Single Response Mode Param	0x01	O
Header	Type	x-bt/InstanceDescription	M
Header	Application Parameters - InstanceID	Varies	M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M
Field	Packet Length	Varies	M
Header	Single Response Mode	0x01	C1
Header	Single Response Mode Param	0x01	C2
Header	Application Parameters - EmailURI	Varies	C3
	- LastUpdate	Varies	C3
Header	Body/End of Body	String with account description	C3

- C1: The Single Response Mode header is mandatory (M) if:
- an SRM header has been received in the previous GET request AND
  - the response code is success (0x90 or 0xA0).
- otherwise excluded (X).
- C2: The Single Response Mode Param header is optional if Single Response Mode is used, otherwise excluded (X).
- C3: These parameters shall be present if the request has been successful.

### 5.8.1 Connection ID

The Connection ID header shall be used to indicate the connection ID according to [6].

### 5.8.2 Type

The type header shall be used to indicate the type of information to be transmitted according to [6].

<x-bt/InstanceDescription>

### 5.8.3 Body/End OfBody

The Body includes a string with the requested user-readable information of account. It shall be represented by a null-terminated UTF-8 text string of at most 200 characters (including the null-termination character).

See GetInstanceInformation function in [6] for further information.

### 5.8.4 Application parameters

#### InstanceID

The InstanceID parameter is the identifier of the CSE-account related to the account information (see also [6]).

#### EmailURI

The EmailURI parameter is a Unicode text string with the SMTP email address related to the CTN account of the CSE. If an email address is not applicable for the related CTN account, this application parameter should be empty.

#### LastUpdate

The LastUpdate parameter is a timestamp of the last update or change of a bCalendar object on the CSE device in format according to Section 3.2.2. The parameter may be used for example to indicate whether there have been changes since the end of the last connection.

## 5.9 SyncCTNAccount function

This function allows the CCE to initiate a synchronization of the corresponding account (instance) with an external server. In particular, this may cause an update of the CSE's repository in case of added, removed, or updated entries. The function follows the general definitions of the GPP SyncInstance function [6].

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	PUT (0x02 or 0x82)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Type	x-bt/Update	M
Header	Body/EndOfBody	Filler-byte "0x30"	M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M

Field	Packet Length	Varies	M
-------	---------------	--------	---

### 5.9.1 Connection ID

The Connection ID header shall be used to indicate the connection ID according to [6].

### 5.9.2 Type

The type header shall be used to indicate the type of information to be transmitted according to [6].

<x-bt/Update>

### 5.9.3 Body/EndOfBody

To avoid PUT with empty body leading to a 'delete', these headers shall contain a filler byte. The value of this byte shall be 0x30 ("0").

## 6 Profile Dependencies

### 6.1 Profile fundamentals

See GPP [6], Section 6.1. No further requirements are stated by CTN.

### 6.2 OBEX

#### 6.2.1 OBEX Services

See GPP [6] for general OBEX requirements (Section 6.2). Within the scope of CTN, the OBEX services CTN Access Service (CAS) and the CTN Notification Service (CNS) are defined (see also Section 4.1).

The CTN features are using the above defined OBEX services in the following way:

Feature	OBEX Service
CTN Account Management	CTN Access Service
CTN Notification	CTN Notification Service
CTN Browsing	CTN Access Service
CTN Downloading	CTN Access Service
CTN Uploading	CTN Access Service
CTN Delete	CTN Access Service
CTN Forward	CTN Access Service

#### 6.2.2 OBEX Operations Used

The following tables list the OBEX operations required by the CTN Profile.

OBEX operations supported in a CTN Access Service session:

OBEX operation	Ability to send ( Client )	Ability to respond ( Server )
	CCE	CSE
Connect	M	M
Disconnect	M	M
Put	M	M
Get	M	M
Abort	M	M

*Table 6-1 OBEX Operations - CAS*

OBEX operations supported in the CTN Notification Service session:

OBEX operation	Ability to send ( Client )	Ability to respond ( Server )
	CSE	CCE
Connect	M	M
Disconnect	M	M
Put	M	M
Abort	M	M

*Table 6.2: OBEX Operations - CNS*

### 6.2.3 OBEX Headers

Table 6.3 lists the OBEX headers required by the CTN Access Profile.

OBEX header	CCE	CSE
Name	M	M
Type	M	M
Description	M	M
Flags	M	M
Body	M	M
End of Body	M	M
Target	M	M
Who	M	M
Connection ID	M	M
Application Parameters	M	M
Single Response Mode	M	M
Single Response Mode Parameters Receive	M	M
Single Response Mode Parameters Send	O	O

*Table 6.3: OBEX Headers*

Note that this profile does not exclude headers that are not listed in Table 6.3. Some implementations may choose to support additional headers, which might enable added value services. Therefore unknown, or unsupported, headers shall always be skipped and ignored.

The following OBEX Target header UUID values shall be used for the OBEX services CAS and CNS:

OBEX service	Target header UUID value
CAS	115f1ec2-2922-11e4-b65d-a6c5e4d22fb7



CNS	45d68d34-2922-11e4-b65d-a6c5e4d22fb7
-----	--------------------------------------

**Table 6.4:** OBEX Target header UUID values

## 6.2.4 Application Parameters Header

The tag IDs used in the Application Parameters header are listed below.

Value	Tag ID	Length	Possible Values
MaxListCount	[6]	[6]	See GPP [6]
ListStartOffset	[6]	[6]	See GPP [6]
NotificationStatus	[6]	[6]	See GPP [6]
InstanceID	[6]	[6]	See GPP [6]
ListingSize	[6]	[6]	See GPP [6]
AcousticAlarmStatus	0x01	1 byte	Bit mask (*): xxxxxxx0 = "off" xxxxxxx1 = "on"
Attachment	0x02	1 byte	Bit mask (*): xxxxxx00 = "on" xxxxxx01 = "off" xxxxxx10 = "select"
Send	0x03	1 byte	Bit mask (*): xxxxxxx0 = "no" xxxxxxx1 = "yes"
FilterPeriodBegin	0x04	variable	string with Begin of filter period See Section 5.3.4
FilterPeriodEnd	0x05	variable	string with End of filter period See Section 5.3.4
ParameterMask	0x06	4 Byte	Bit mask (*): See Section 5.3.4
StatusIndicator	0x07	1 byte	Bit mask (*): xxxxxx00 = "pStatus" xxxxxx01 = "alarmStatus" xxxxxx10 = "sendStatus" xxxxxx11 = "deletedStatus"
StatusValue	0x08	1 byte	Bit mask (*): xxxx0000 = "no" xxxx0001 = "yes" xxxx0010 = "postpone" xxxx0011 = "tentative" xxxx0100 = "needs-action" xxxx0101 = "accepted" xxxx0110 = "declined"

Value	Tag ID	Length	Possible Values
			xxxx0111 = "delegated" xxxx1000 = "completed" xxxx1001 = "in-progress"
PostponeVal	0x09	4 byte	Positive integer, 0...n
EmailURI	0x0A	variable	Text
CSETime	0x0B	variable	string with current local time basis and UTC-offset of the CSE See Section 5.3
Recurrent	0x0C	1 byte	Bit mask (*): xxxxxxx0 = "no" xxxxxxx1 = "yes"
AttachID	0x0D	1 byte	Positive integer, 1...n
LastUpdate	0x0E	variable	string with time of last update See Section 5.8.4

**Table 6.5: Tag IDs**

(\*) the bits marked by 'x' shall be set to zero; all other values are reserved for future use

All of the Application Parameter header values use big-endian byte ordering.

### 6.2.5 OBEX headers in Multi-Packet Responses

See GPP [6], Section 6.2.6.

### 6.2.6 OBEX Error Codes

See GPP [6], Section 7.2.7.

## 6.3 Link Control (LC) Interoperability Requirements

See GPP [6], Section 6.3. No further requirements are stated by CTN.

## 6.4 Generic Access Profile

See GPP [6], Section 6.4. No further requirements are stated by CTN.

## 6.5 Modes

See GPP [6], Section 6.5. No further requirements are stated by CTN.

## 6.6 Security aspects

See GPP [6], Section 6.6. No further requirements are stated by CTN.

## 6.7 Idle mode procedures

See GPP [6], Section 6.7. No further requirements are stated by CTN.

V1.0.0

## 7 Service Discovery

### 7.1 SDP Interoperability Requirements

The following service records are defined for the CTN Profile. There shall be one service record per CCE device (CTN Notification Service, server role) and on the CSE device, one for each CAS instance (CTN Access Service, server role). See also GPP [6].

### 7.2 SDP record for the CTN Access Service on the CSE device

Item	Definition	Type	Value	Status	Default
ServiceClassID List				1. M	
2. ServiceClass #0		3. U UI D	4. CTN Access Service	5. M	
6. Protocol Descriptor List				7. M	
8. Protocol #0		9. U UI D	10. L2CAP	11. M	
12. Param #0		13. Ui nt 16	14. Varies <sup>1</sup>	15. M	
16. Protocol #1		17. U UI D	18. OBEX	19. M	
20. ServiceName		21. Str ing	22. Service provider defined	23. M	24. Acco unt name
25. Bluetooth Profile Descriptor List				26. M	
27. Profile #0	28. Suppor ted Profiles	29. U UI D	30. Calendar Tasks and Notes profile	31. M	
32. Param #0		33. Ui nt 16	34. V1.0	35. M	36. 0x01 00
37. CTN InstanceID		38. Ui nt		39. M	

V1.0.0

		8			
40. CTN Supported Features		41. Uint32	42. Bit 0 = CTN Account Management Feature  Bit 1 = CTN Notification Feature  Bit 2 = CTN Browsing Feature  Bit 3 = CTN Downloading Feature  Bit 4 = CTN Uploading Feature  Bit 5 = CTN Delete Feature  Bit 6 = CTN Forward Feature  Bit 7 ~ 31 Reserved <sup>2</sup>	43. M	

<sup>1</sup> L2CAP PSM values are dynamic.

<sup>2</sup> Reserved bits shall be set to 0 and ignored on reception.

For CTN SDP UUID values, please refer to the Bluetooth Assigned Numbers section of the Bluetooth website [12].

### 7.3 SDP record for the CTN Notification Service on the CCE device

Item	Definition	Type	Value	Status	Default
ServiceClassID List				44. M	
45. ServiceClass #0		46. UUID	47. CTN Notification Service	48. M	
49. Protocol Descriptor List				50. M	
51. Protocol #0		52. UUID	53. L2CAP	54. M	
55. Param #0		56. Uint16	57. Varies <sup>1</sup>	58. M	

59. Protocol #1		60. UUID	61. OBEX	62. M	
63. ServiceName		64. String	65. Service provider defined	66. M	67. CNS name
68. Bluetooth Profile Descriptor List				69. M	
70. Profile #0	71. Supported Profiles	72. UUID	73. Calendar Tasks and Notes profile	74. M	
75. Param #0		76. Uint16	77. V1.0	78. M	79. 0x0100
80. CTN SupportedFeatures		81. Unit32	82. Bit 0 = CTN Account Management Feature  Bit 1 = CTN Notification Feature  Bit 2 = CTN Browsing Feature  Bit 3 = CTN Downloading Feature  Bit 4 = CTN Uploading Feature  Bit 5 = CTN Delete Feature  Bit 6 = CTN Forward Feature  Bit 7 ~ 31 Reserved <sup>2</sup>	83. M	

<sup>1</sup> L2CAP PSM values are dynamic.

<sup>2</sup> Reserved bits shall be set to 0 and ignored on reception.

For CTN SDP UUID values please refer to the Bluetooth Assigned Numbers section of the Bluetooth website [\[12\]](#).

## 8 Acronyms and Abbreviations

Abbreviation or Acronym	Meaning
CAS	CTN Access Service
CCE	CTN Client Equipment
CNS	CTN Notification Service
CSE	CTN Server Equipment
FRD	Feature Requirements Document
GAP	Generic Access Profile
GPP	Generic PIM Profile;
GOEP	Generic Object Exchange Profile
L2CAP	Logical Link Control and Adaptation Protocol
LMP	Link Manager Protocol
MRD	Market Requirements Document
MSC	Message Sequence Chart
OBEX	Object Exchange
PIM	Personal Information Management
WG	Working Group

**Table 8.1:** *Abbreviations and Acronyms*

## 9 References

---

- [1] Specification of the Bluetooth System, Core Package Version 2.1 + EDR or later
- [2] Bluetooth Specification [vol0]; Part E IEEE Language
- [3] IrDA Object Exchange Protocol OBEX™ with Published Errata, Version 1.5, iAnywhere Inc, Extended systems Inc. and Microsoft Corporation
- [4] Generic Object Exchange Profile v2.0 or later, Bluetooth SIG
- [5] Generic Access Profile Specification, latest version applies, Bluetooth SIG
- [6] Generic PIM Profile Specification, latest version applies, Bluetooth SIG
- [7] Feature Requirements Document, Calendar Tasks and Notes Profile V1.0, 23 August 2011
- [8] RFC 5545, Internet Calendaring and Scheduling Core Object Specification (iCalendar)
- [9] vCard The Electronic Business Card, version 2.1, September 18th
- [10] Extensible Markup Language (XML) 1.0 (Fourth Edition) W3C Recommendation 29 September 2006, Definitions, <http://www.w3.org/TR/xml/>
- [11] Multipurpose Internet Mail Extensions (MIME), Part 2, Media Types, RFC 2046
- [12] Bluetooth SIG Assigned Numbers

V1.0.0