



# Bluetooth Intercom Profile Software Interface Specification

Tality Part No: T-SS-BT-3008-100



## Proprietary Notice

© 2001 Tality Corporation. All rights reserved.

In the U.S. and numerous other countries, Tality and the Tality logo are trademarks of Tality Corporation. Bluetooth and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc, USA and licensed to Tality Corporation. Use of the Specification by anyone who is not a member of Bluetooth SIG or a party to any Early Adopters Agreement (such person or party, a “Member”), is prohibited. The legal rights and obligation of each Member are governed by their applicable Membership Agreement, Early Adopter Agreement or Promoters Agreement. No license, expressed or implied, by estoppel or otherwise, to any intellectual property rights are granted herein

All other products or services mentioned herein may be trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in this document may be adapted or reproduced in any material form except with the prior written permission of the copyright owner.

The product described in this document is subject to continuous developments and improvements and is supplied “AS IS”. All warranties implied or expressed including but not limited to implied warranties or merchantability, or fitness for purpose, are excluded. Tality Corporation shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Tality products are not authorized for use as critical components in life support devices or systems without the express written approval of an authorized officer of Tality Corporation. As used herein:

1. Life support devices or systems are devices or systems that are:
  - intended for surgical implant into the body or
  - support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labelling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system or system whose failure to perform can reasonably be expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

## Table of Contents

1.	Introduction .....	1
2.	Intercom Profile Interface .....	2
	2.1. Registration Services .....	2
	2.1.1. Register .....	2
	2.1.2. De-Register .....	3
	2.1.3. Create Service Record .....	3
	2.1.4. Delete Service Record .....	4
	2.2. Call Control Procedures .....	4
	2.2.1. Intercom Profile Version Check .....	4
	2.2.2. Call .....	5
	2.2.3. Call Clearing .....	5
	2.2.4. Alerting .....	6
	2.2.5. Answer .....	6
	2.3. Bearer Connection .....	6
	2.3.1. Bearer Channel Connected .....	6
	2.3.2. Bearer Channel Disconnected .....	7
3.	ICP Message Sequence Charts .....	8
	3.1. Registration .....	8
	3.2. Inquiry .....	8
	3.3. Service Check .....	9
	3.4. Call Request .....	9
	3.5. Connection Alert .....	9
	3.6. Call Clearing .....	10
4.	ICP Parameter Definition .....	11
5.	ICP Configurable Definitions .....	12
6.	ICP Result Codes .....	13
7.	Reference Documents .....	14
8.	Contact Information .....	15
	8.1. Tality .....	15
	8.1.1. What can Tality do for you? .....	15
	8.1.2. About Tality .....	15

# 1. Introduction

This document defines the interface that is supported by Tality's Intercom Profile (ICP) and implements the Bluetooth™ intercom profile as specified in Bluetooth Specification, Version 1.1 (Ref 1).

Application software controlling a Bluetooth intercom device uses the services of Tality's ICP to handle all the Bluetooth specific aspects of its operation. In turn, Tality's ICP provides its features by using the services of the Tality Bluetooth Protocol Software Licensed Design.

Tality's ICP is strongly dependent up the TCS component of the licensed design and the features that the interface provides are set out in the Software Requirements Specification (Ref 2).

Tality profile software operates in Tality's Bluetooth software kernel environment using a pure message passing mechanism. This kernel is designed to be small in size for efficiency in embedded systems and has the following features:

- portable across different processors
- low ROM and RAM usage
- simple in operation
- directly handles SDL features such as timers, finite state machines (FSMs) and message-based events to simplify the coding of SDL diagrams
- contains buffer and message management

It uses Tality's SDP implementation as documented in SDP Services Interface Specification (Ref 3). It also uses Tality's TCS implementation as documented in TCP Lean Services Interface Specification (Ref 4).

This document is for public information and is an abridged version of the software specification for this design. For further details, please contact Tality Corporation.

## 2. Intercom Profile Interface

Interfaces exported by the ICP are shown in tables below. Each column represents the request, confirm, indication and response messages of the interface. If a column is greyed out, then those request, confirm, indication or response message is not supplied. Each row represents the interface parameter that may or may not required by the message. Ticks indicate which parameters are available in each message. For example, ICP\_REGISTER interface has request and confirm messages, but with no indication and response messages.

Some request messages require identity of the originator so that confirms and other related messages may be sent to the correct task. This is indicated by an *italicised parameter originator*. This parameter does not appear in the payload portion of the message. It is stored in the originator field of the kernel message header. A kernel function (vKNL\_SET\_ORIGINATOR) is available to set this field before the message is sent.

### 2.1. Registration Services

The services are used by applications for two principal purposes:

- Registration with the ICP to accept or originate intercom calls
- Creation of a suitable service record in the service discovery protocol (SDP) database of the local device.

#### 2.1.1. Register

An application uses this service to register with the ICP and can only register once with the ICP. Only one application can be registered with the ICP at any given time.

Until an application has registered with the ICP, any incoming calls are rejected automatically. An application must be registered with the ICP before it can initiate an intercom call.

If the link type is ICP\_BC\_NONE, no physical bearer channel is established by TCS. In this case, the application has the responsibility of establishing a bearer channel between two Bluetooth devices.

The application puts its task reference into the related message area (sFrom) before sending this message. The ICP uses this task reference for all response messages afterwards.

<b>Service</b>	ICP_REGISTER				
<b>Function</b>	Register with the intercom profile to accept or originate intercom calls				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---	✓	✓		
<i>Originator</i>	<i>Task ID</i>	✓			
Link Type	Link type field should be either ICP_BC_SCO or ICP_BC_NONE	✓			
Status	ICP_STATUS_OK, ICP_STATUS_NO_RESOURCE, ICP_STATUS_ALREADY_REGISTERED, ICP_STATUS_UNSUPPORTED_LINK_TYPE		✓		

### 2.1.2. De-Register

An application uses this service to relinquish its registration with the ICP and can only de-register if it has previously registered successfully with the ICP. An application can only de-register if the ICP has no calls in progress and clears any calls before it de-registers.

<b>Service</b>	ICP_DEREGISTER				
<b>Function</b>	Relinquish registration with the intercom profile.				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---	✓	✓		
Status	ICP_STATUS_OK, ICP_STATUS_DELETE_SR_FIRST, ICP_STATUS_DEVICE_BUSY		✓		

### 2.1.3. Create Service Record

An application uses this service to create a service record in the local SDP database. The format of the service record conforms to that described in the Software Requirement Specification (Ref 2).

It is not mandatory the application uses this service to populate the SDP database as the application can also use the SDP interface directly to create the service record. In this case, it is the responsibility of the application to ensure that the format of the data in the service record is correct and conforms to the features provided by the ICP.

<b>Service</b>	ICP_CREATE_SR				
<b>Function</b>	Create a Service Record in the SDP database.				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---	✓	✓		
Length	Length of service name	✓			
Service Name	The byte string to appear in the service name field of the SDP service record	✓			
Status	ICP_STATUS_OK, ICP_STATUS_SR_EXISTS, ICP_STATUS_INVALID_LENGTH, ICP_STATUS_SDP_ERROR, ICP_STATUS_DEVICE_BUSY		✓		

### 2.1.4. Delete Service Record

An application uses this service to delete the ICP service record from the local SDP database. It is not mandatory that the application uses this service to delete the service record from the SDP database as the application can use the SDP interface directly.

<b>Service</b>	ICP_DELETE_SR				
<b>Function</b>	Delete the ICP service record from the SDP database.				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---	✓	✓		
Status	ICP_STATUS_OK, ICP_STATUS_NO_SR, ICP_STATUS_SDP_ERROR, ICP_STATUS_DEVICE_BUSY		✓		

## 2.2. Call Control Procedures

### 2.2.1. Intercom Profile Version Check

An application uses this service to check the ICP version of the remote Bluetooth device. The version is checked to define if there are any features unsupported on the remote device.

The application performs an inquiry to find the intercom devices in range. Connect handle and SR handle parameters are obtained after this operation. If there is a version difference between local and remote profile, the local profile does not take any action. The application has the responsibility to decide which features are used depending on the version of the remote profile.

<b>Service</b>	ICP_SERVICE_CHECK				
<b>Function</b>	Checks the ICP version number on the remote Bluetooth device				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---	✓	✓		
Connect Handle	SDP connection handle to remote device.	✓			
SR Handle	Service record handle of service to check.	✓			
Version	Profile version (0x0100 indicates version 1.0)		✓		
Status	ICP_STATUS_OK, ICP_STATUS_SDP_ERROR, ICP_STATUS_DEVICE_BUSY, ICP_STATUS_NOT_COMPATIBLE		✓		

### 2.2.2. Call

An application uses this service to establish an intercom call with another Bluetooth device. If an application has registered with the ICP, then the ICP uses this service to notify the registered application of incoming intercom calls.

BD\_ADDR should be obtained before ICP\_CALL using inquiry process.

<b>Service</b>	ICP_CALL				
<b>Function</b>	Establishment of an intercom call				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---	✓	✓	✓	✓
BD_ADDR	Bluetooth address	✓	✓	✓	✓
Cause	Extra information in the case of call failure or rejection.		✓		✓
Status	ICP_STATUS_OK, ICP_STATUS_TCS_ERROR, ICP_STATUS_DEVICE_BUSY		✓		✓

### 2.2.3. Call Clearing

An application uses this service to clear an existing intercom call. If an application has registered with the ICP and a call is in progress, the ICP uses this service to notify the registered application that it wishes to clear the call.

ICP\_CLEAR indication is transmitted to both communicating devices after any line break or link loss. If the device gets ICP\_CLEAR message with the ICP\_STATUS\_CLEAR\_PENDING status, then it means that the clear operation has started but not finished yet. The device cannot handle any outgoing calls or other ICP interface operations. The application waits until receiving another ICP\_CLEAR\_IND with the status ICP\_STATUS\_OK.

<b>Service</b>	ICP_CLEAR				
<b>Function</b>	Clearing of an intercom call				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---	✓	✓	✓	✓
Cause	Values defined in Section 4.3.3 of ICP Software Requirement Specification	✓		✓	
Status	ICP_STATUS_OK, ICP_STATUS_TCS_ERROR, ICP_STATUS_DEVICE_BUSY, ICP_STATUS_CLEAR_PENDING		✓	✓	✓

### 2.2.4. Alerting

An application uses this service to receive notification that the remote device has received the call and the called party (remote user) has been alerted of an incoming call. The outgoing side then changes its state to call delivered. The called delivered state exists for an outgoing call when the sender has received an indication that the receiver has received the alert, but has not yet answered the call.

<b>Service</b>	ICP_ALERTING				
<b>Function</b>	Alerting message from the remote Bluetooth device				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---			✓	

### 2.2.5. Answer

The incoming application uses this service to inform ICP that the user has answered the call.

<b>Service</b>	ICP_ANSWER_EVT				
<b>Function</b>	Answer message from the remote Bluetooth device				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---			✓	

## 2.3. Bearer Connection

An application needs to know the exact time when the speech path was established and when it can start to use the bearer channel. TCS places the responsibility for the control of these external channels, or circuits with the client. Therefore it is necessary for ICP to indicate to the application when to connect to or disconnect from the external interface of the bearer channel after it has exchanged the channel information with TCP.

The application may select link type as ICP\_BC\_NONE when registering with the ICP. In this case, the BC\_ID has no direct relationship with the physical bearer channel ID, but the application receives both indication messages and the ICP waits for the related responses.

### 2.3.1. Bearer Channel Connected

The ICP transmits this message to the application after TCS establishes the bearer channel connection and has sent an indication to the ICP. Even if the link type is set to ICP\_BC\_NONE, this message is transmitted to the application. In this case, the bearer channel ID has no direct link with a physical bearer channel and the application has the responsibility of establishing a bearer channel between two devices.

<b>Service</b>	ICP_BC_CONNECT				
<b>Function</b>	Bearer channel connection				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---			✓	✓
BC_ID	Bearer channel ID. Identifier for an existing bearer channel			✓	

### 2.3.2. Bearer Channel Disconnected

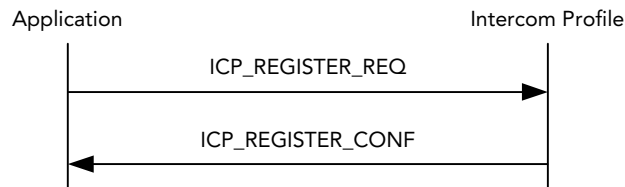
An application uses this service to receive indication message of the bearer channel disconnection. The ICP also transmits this indication message to the application if a problem occurs with the bearer channel connection or with the signalling link and the connection has broken. The indication message can be transmitted at any time during the call active state and informs of any exceptional interruptions on the channel. This message is also transmitted during the normal call clearing process.

<b>Service</b>	ICP_BC_DISCONNECT				
<b>Function</b>	Bearer channel disconnection				
<b>Parameter</b>	<b>Description</b>	<b>Request</b>	<b>Confirm</b>	<b>Indication</b>	<b>Response</b>
Present	---				
BC_ID	Bearer channel ID. Identifier for an existing bearer channel				

### 3. ICP Message Sequence Charts

#### 3.1. Registration

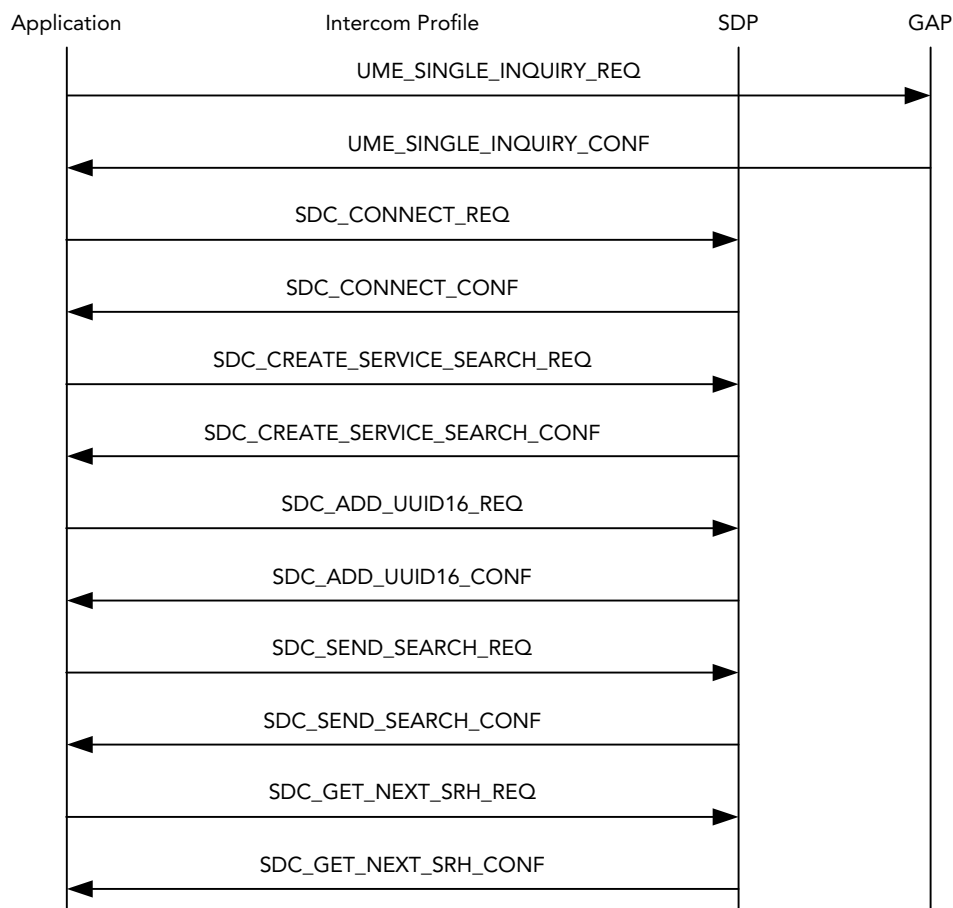
After registration has finished, the application is able to use all other ICP services.



#### 3.2. Inquiry

After UME\_SINGLE\_INQUIRY\_CONF, the application finds any devices that are in range using GAP.

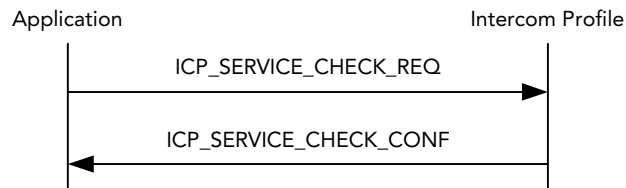
After SDC\_CONNECT\_CONF, the application starts a service search to find if any device supports the ICP.



The application repeats the request for a service record handle until the SDP client replies with the status indicating that there are no more matching service records.

### 3.3. Service Check

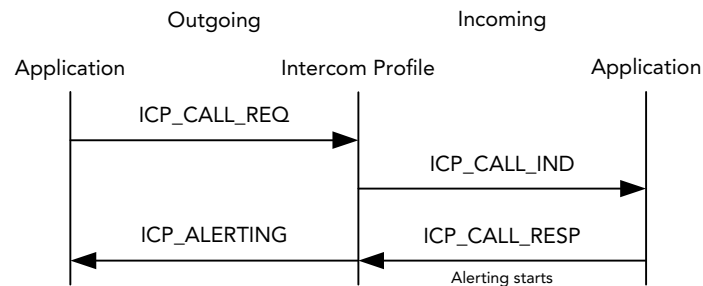
The ICP talks to the SDC to find the remote profile's version and returns it to application.



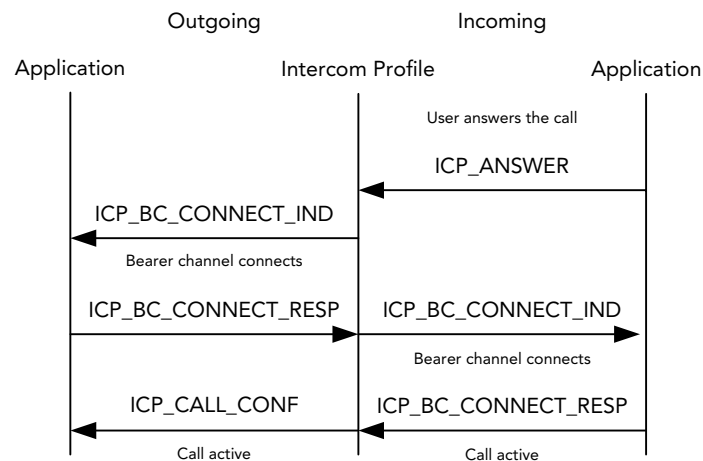
At this point the application knows the remote device's ICP Version and can decide to limit the functionality if necessary or not to connect to that specific version.

### 3.4. Call Request

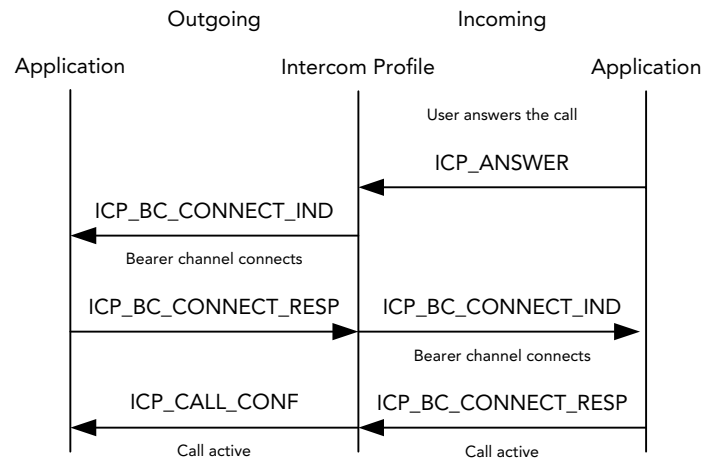
The application finds any Bluetooth devices in range. The user starts the call process choosing which device to use.



### 3.5. Connection Alert



### 3.6. Call Clearing



## 4. ICP Parameter Definition

Parameter Name	Description	Type/Length
Task Ref	The licensed design kernel task identification. Composed of a task ID and an instance number. Used to identify a particular task within the protocol stack.	K_TASKREF
BD_ADDR	Address of the end point unit	Unsigned integer (48 bits)
Link Type	Link type field should be either CTP_BC_SCO or CTP_BC_NONE] as defined on Section 4.3.1 of the Bluetooth Specification (Ref 1).	ICP_LINKTYPE
BC_ID	Bearer channel identifier	K_INSTANCE
Length	Length of service name	Unsigned integer (8 bits)
Service Name	The byte string to appear in the service name field of the SDP service record	u8 string
Version	Profile version (0x0100 indicates version 1.0)	Unsigned integer (16 bits)
Connect Handle	SDP connection handle to remote device	Unsigned integer (16 bits)
SR Handle	Service record handle of service to check	Unsigned integer (32 bits)
Cause	Extra information in the case of call failure. Values are defined in Section 4.3.3 of the Software Requirements Specification (Ref 2).	Unsigned integer (8 bits)
Status	Status code indicating success or one of several error codes	Unsigned integer (16 bits)

## 5. ICP Configurable Definitions

Definition	Value	Filename	Description
ICP_TIMER_TIC100_TIMEOUT_PERIOD	10 seconds	icp.h	Intercom timer TIC100 time out period.
ICP_TIMER_DISCONNECT_TIMEOUT_PERIOD	10 seconds	icp.h	The timer is started when waiting for disconnect indication.
ICP_TIMER_DISCONNECT_REQ_TIMEOUT_PERIOD	1 seconds	icp.h	The timer is a delay timer started on incoming side before sending a disconnect request to stop L2CAP disconnect collision on tests
ICP_MAXIMUM_SERVICE_NAME_LENGTH	128 bytes	icp.h	Max service name length allowed by ICP
ICP_LOCAL_VERSION	0x0100	icp.h	Profile version. Used when comparing with the remote profile's version
ICP_TCS_CALL_CLASS_INTERCOM	0x01	icp.h	Intercom call class as documented in the Bluetooth specification.
ICP_PT_UI11_VALUE	0x27	cfgbprof.h	Packet type and user information layer 1 values. Can be changed at design time. 0x27 means Packet Type = HV3 User Information Layer 1 = CVSD

## 6. ICP Result Codes

Status Code	Description
ICP_STATUS_OK	Service request has been successful, or client accepts indication message.
ICP_STATUS_NO_RESOURCE	A service request could not be completed, or a response to a service indication could not be given, due to resource exhaustion.
ICP_STATUS_ALREADY_REGISTERED	Only one application can register at any given time and an application cannot register two times.
ICP_STATUS_UNSUPPORTED_LINK_TYPE	Link type cannot have any value other than ICP_BC_SCO or ICP_BC_NONE.
ICP_STATUS_DELETE_SR_FIRST	If the application created a service record using the ICP service, first it should delete the service record before it deregisters.
ICP_STATUS_INVALID_LENGTH	Service name cannot be longer than 128 bytes.
ICP_STATUS_SR_EXISTS	The application can not create another SR before deleting the previous one. If the application wants to change the name first, it should delete the previous SR and then create a new name.
ICP_STATUS_NO_SR	The application can not delete a service record if it has not used the service interface to create the record.
ICP_STATUS_SDP_ERROR	ICP got an unexpected error when using the SDP interface.
ICP_STATUS_TCS_ERROR	A service request could not be completed, or a response to a service indication could not be given due to an error at TCS layer.
ICP_STATUS_NOT_COMPATIBLE	Remote device's profile version is not compatible with the local device.
ICP_STATUS_CLEAR_PENDING	Clear operation has started but not finished yet. The device cannot handle any outgoing calls or other ICP interface operations. The application has to wait until receiving another ICP_CLEAR_IND with the status ICP_STATUS_OK.
ICP_STATUS_DEVICE_BUSY	ICP cannot handle the request since a call is active or just about to be active.

## 7. Reference Documents

1. Bluetooth Core Specification, Version 1.1, Bluetooth SIG Document 1.C.47/1.1
2. Intercom Profile Software Requirement Specification, Tality Document S10973, Issue 1.0
3. SDP Services Interface Specification, Tality Document S10587, Issue 0.4.
4. TCP Lean Services Interface Specification, Tality Document S10962, Issue 1.3.

## 8. Contact Information

### 8.1. Tality

#### 8.1.1. What can Tality do for you?

Tality undertakes all aspects of communications product design, including complete system and sub-system design, and both hardware and software component design. The company focuses primarily on network infrastructure, wireless infrastructure, broadband access devices, and consumer communications and infotainment products. Tality's design services span product conceptualisation through implementation for production manufacturing, enabling our customers to get better products to market faster.

#### 8.1.2. About Tality

Tality Corporation, a subsidiary of Cadence Design Systems, Inc. (NYSE:CDN), is the world's largest electronic product development outsourcing provider. Leading and emerging technology companies around the globe leverage Tality's engineering services and intellectual property for the design of complex electronic systems and integrated circuits. Tality is headquartered in San Jose, California. For more information, contact us at +44 (0) 1506 595000, or e-mail to [bluetooth@tality.com](mailto:bluetooth@tality.com) or visit us on the World Wide Web at: <http://www.tality.com>.