Enabling Bluetooth Limited Inquiry (LIAC) using Symbian C++

Overview

This code snippet shows how to add Limited Inquiry Access Code (LIAC) support to the S60 Platform: Bluetooth Point-to Multipoint Example. A separate patch package is given at the end of the article.

In the Bluetooth Specification there are two different inquiry access codes for different use cases: Limited Inquiry Access Code (LIAC) and Generic Inquiry Access Code (GIAC). **The default inquiry method is Generic Inquiry (GIAC)** that finds all the visible devices. **By default the listening devices are of type Discoverable, which means that they are scanning with GIAC.**

You can set both the inquiry and the scanning method into LIAC mode. When making a device inquiry with LIAC, devices are found considerably faster because only those devices which are in LIAC mode are found and other normal discoverable devices are filtered out. Because fewer devices are found, this also speeds up the subsequent service discovery procedure performed on discovered devices.

Limited Inquiry (LIAC) is useful in cases where an application is running on two or more devices and connection is desired only between these devices. Devices scanning with LIAC can be found by devices inquiring with LIAC or GIAC (LIAC inquiries are prioritized).

Devices inquiring with LIAC can only find devices scanning with LIAC.

Note: After the changes described in this code snippet, the application needs to be signed via the Open Signed process. More information about Open Signed is available at Symbian Signed.

Preconditions

Limited Device Status can only be enabled on S60 3rd Edition or higher.

MMP file

The following capabilities are required:

CAPABILITY LocalServices
CAPABILITY NetworkControl
CAPABILITY WriteDeviceData

Header file: BluetoothPMPExampleEngine.h

Add these lines to the existing header (`BluetoothPMPExampleEngine.h`):

```
#include <e32property.h> // RProperty
RProperty iProperty;
```

Source file: BluetoothPMPExampleEngine.cpp

From S60 3rd Edition onwards, the discoverability is set via the Bluetooth Publish and Subscribe API using a constant `KPropertyKeyBluetoothSetLimitedDiscoverableStatus`.

To set the listening device to Limited-Discoverable mode, add this code to the end of `CBluetoothPMPExampleEngine::ConstructL()`:

```
// Define Limited-Discoverable mode
TInt attErr = iProperty.Define(KPropertyUidBluetoothControlCategory,
    KPropertyKeyBluetoothSetLimitedDiscoverableStatus,RProperty::EInt);
if (attErr != KErrNone && attErr != KErrAlreadyExists)
    {
    User::Leave(attErr);
    }

// Turn Limited-Discoverable mode on
User::LeaveIfError(iProperty.Set(KPropertyUidBluetoothControlCategory,
    KPropertyKeyBluetoothSetLimitedDiscoverableStatus, ETrue));
```

Remember to close RProperty in the destructor (`CBluetoothPMPExampleEngine::~CBluetoothPMPExampleEngine()`):

```
iProperty.Close();
```


(C) Copyright Nokia 2013. All rights reserved.

Header file: DeviceDiscoverer.h

Add these lines to the existing header (`DeviceDiscoverer.h`):

```
#include <bttypes.h> // KLIAC constant
```

Source file: DeviceDiscoverer.cpp

To change the discovering device from Generic Inquiry to Limited Inquiry, add the following to the `CDeviceDiscoverer::DiscoverDevicesL()` function:

```
iAddr.SetIAC(KLIAC);
```

Postconditions

S60 Platform: Bluetooth Point-to-Multipoint Example has been changed to support Limited Inquiry.

Patch

File:BtPMPEx LIAC patch.zip: Patch against S60 Platform: Bluetooth Point-to Multipoint Example

The patch package contains:

- LIAC.patch file that can be applied to the S60 Platform: Bluetooth Point-to Multipoint Example
- LIAC.patch.html file that illustrates all the changes made to the original example application

 (C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.