Archived:Discovering Bluetooth devices using Symbian C++

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Overview

This article shows how to search Bluetooth devices with `RHostResolver`.

`RHostResolver` provides an interface to host name resolution services, such as DNS, that may be provided by certain protocol modules.

The interface provides functions to access the following facilities:

- Obtaining names from addresses
- Obtaining addresses from names
- Getting and setting local host name

Before using any service, a connection to a socket server session must be established.

Each function is available in both synchronous and asynchronous versions.

MMP file

The following capabilities and libraries are required:

```
CAPABILITY LocalServices
LIBRARY esock.lib
LIBRARY bluetooth.lib
```

Header file

```
#include <es_sock.h>
#include <bt_sock.h>

RSocketServ iSocketServ;
RHostResolver iResolver;
TInquirySockAddr iAddr;
```

Source file

`CMyDeviceDiscoverer` must be an active object (derived from CActive).

```
_LIT(KBTLinkManagerTxt,"BTLinkManager");


void CMyDeviceDiscoverer::ConstructL()
    {
    ...
    // Get socket server session
    User::LeaveIfError(iSocketServ.Connect());
    ...
    }

void CMyDeviceDiscoverer::DiscoverDevicesL()
    {
    if (!IsActive())
        {
        // Load protocol for discovery
        TProtocolDesc pdesc;
        User::LeaveIfError(
            iSocketServ.FindProtocol(KBTLinkManagerTxt(), pdesc));

        // Initialize host resolver
        iResolver.Close();
        User::LeaveIfError(
            iResolver.Open(iSocketServ, pdesc.iAddrFamily, pdesc.iProtocol));

        // Start device discovery by invoking remote address lookup
        iAddr.SetIAC( KGIAC );
        iAddr.SetAction(KHostResInquiry|KHostResName|KHostResIgnoreCache);
        iResolver.GetByAddress(iAddr, iEntry, iStatus);
        SetActive();
        }
    else
        {
        User::Leave(KErrNotReady);
        }
    }
```


(C) Copyright Nokia 2013. All rights reserved.

```
void CMyDeviceDiscoverer::RunL()
    {
    // RHostResolver.GetByAddress() has completed, process results

    if ( iStatus!=KErrNone )
        {
        // All devices find
        }
    else
        {
        // New device data entry

        // Device name
        iEntry().iName;

        // Device address
        static_cast<TBTSockAddr>(iEntry().iAddr).BTAddr();

        // Get next discovered device
        iResolver.Next(iEntry, iStatus);
        SetActive();
        }
    }

void CMyDeviceDiscoverer::DoCancel()
    {
    iResolver.Cancel();
    }

CMyDeviceDiscoverer::~CMyDeviceDiscoverer()
    {
    Cancel();
    iResolver.Close();
    }
```

Postconditions

`CMyDeviceDiscoverer` finds existing Bluetooth devices.

See also

Archived:Discovering Bluetooth services using Symbian C++

Advertising Bluetooth services using Symbian C++

Archived:Establishing a Bluetooth connection using Symbian C++

Archived:RHostResolver and redundant display of access point selection dialog (Known Issue)

Example application

This code snippet has been used in the S60 Platform: Bluetooth Point-to-multipoint Example application.

(C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.