Making Bluetooth connection by touching two NFC enabled (C7) phones

Introduction

In this code snippet, we are going to show and explain how we can use Symbian API to connect with each other by just touching two NFC enabled Symbian devices (two C7 for example). When our application has complex setting then we can pass the settings to other devices just by touching and making a Bluetooth connection. This can be used when we need to develop multi-player Bluetooth game over Bluetooth where you just touch and play.

How it works

In this code snippet, we can use the traditional way to find Bluetooth and connect to it or by touching each other. When we touch two devices then we get the Bluetooth address according to this article (How to find Bluetooth device name by touching two NFC devices. This address is used find the target service offered by other device. Following code snippet shows how we can use the found BT address to find service.

```
void CMessageClient::ConnectBTL(const TDesC8 & aBTAddress)
{
    if(aBTAddress.Length() != 12)
     {
      User::Leave( KErrArgument );
      }
      TBuf<4> upper(0);
      TBuf<8> lower(0);
      TUint32 up = 0;
      TUint32 low = 0;
      const TUint8 *ad = aBTAddress.Ptr();
      for(TInt i = 0; i < 4; i++)
      {
       upper.Append(static_cast <TChar >(*ad));
       ad++;
       }
        for(TInt i = 0; i < 8; i++)
       {
       lower.Append(static_cast <TChar >(*ad));
       ad++;
     }
     TLex lexer;
     lexer.Assign(upper);
     lexer.BoundedVal(up,EHex, 0xFFFFFFFF);
     lexer.Assign(lower);
     lexer.BoundedVal(low, EHex, 0xFFFFFFFF);
     TBTDevAddr devAddr(MAKE_TINT64(up, low));
     iState = EGettingService;
     iStatus = KRequestPending; // this means that the RunL can not be called until
if(!IsActive())
    {
     iServiceSearcher->FindServiceL(iStatus, devAddr); // devAddr must be saved for
future reference.
     SetActive();
     ...
```


(C) Copyright Nokia 2014. All rights reserved.

In traditional way we get the BT address from the notifier response (iResponse().BDAddr()). Since we are not using notifier here, we need to save the Bluetooth address and pass it to CSdpAgent.

```
// delete any existing agent and search pattern
void CBTServiceSearcher::FindServiceL(TRequestStatus& aObserverRequestStatus, const
TBTDevAddr & aDevice)
{
iHasFoundService = EFalse;
iWithNFC = ETrue;  // By NFC or not?
iTBTDevAddr = aDevice;

// delete any existing agent and search patterndelete iSdpSearchPattern;
iSdpSearchPattern = NULL;

delete iAgent;
iAgent = NULL;


iAgent = CSdpAgent::NewL(*this, iTBTDevAddr);

iSdpSearchPattern = CSdpSearchPattern::NewL();

iSdpSearchPattern->AddL(ServiceClass());
// return code is the position in the list that the UUID is inserted at delete
iSdpSearchPattern;
iSdpSearchPattern = NULL;
delete iAgent;
iAgent = NULL;

iAgent = CSdpAgent::NewL(*this, iTBTDevAddr);

iSdpSearchPattern = CSdpSearchPattern::NewL();

iSdpSearchPattern->AddL(ServiceClass());

iStatusObserver = &aObserverRequestStatus;
iAgent->NextRecordRequestL();
}
```

How to use this example

There is a sis file in the sis folder. That can be signed and install in C7 devices. If you run the application and then touch two devices then these are automatically connected. Now you can send message from client to server or server to client as usual way. This example application is based on Bluetooth serial profile. NFC enabled headset and C7 work on the same principle.

Example Code

The following code was used to test and pass message from one phone to other over Bluetooth with two C7 devices. Download the example for C7: File:NFCBtChatSymbian.zip

 (C) Copyright Nokia 2014. All rights reserved.

 (C) Copyright Nokia 2014. All rights reserved.