Archived:Bluetooth Engine API for S60 3rd Edition FP2

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*}}.

ID	CS001064	**Creation date**	July 7, 2008
Platform	S60 3rd Edition, FP2	**Tested on devices**	Nokia 6210 Navigator
Category	Symbian C++	**Subcategory**	Bluetooth

Note: :This API is not part of the public SDK. It can be found in the SDK API Plug-in.

Purpose

The main purpose of the Bluetooth Engine is to provide a higher-level abstraction API for Bluetooth applications and services. The Bluetooth Engine API provides access to Bluetooth settings, Bluetooth device management, and Bluetooth connection settings.

The Bluetooth API for S60 3rd Edition, FP1 and older devices is available here. Note that there is a compatibility break because the Bluetooth architecture has changed starting from S60 3rd Edition, Feature Pack 2.

Header files in the package:

```
#include <btengsettings.h>
#include <bt_sock.h>
#include <btdevice.h>
#include <btengconnman.h>
#include <btengconstants.h>
#include <btengdevman.h>
#include <btengdiscovery.h>
#include <btengdomaincrkeys.h>
#include <btmanclient.h>
#include <btsdp.h>
```

Libraries in the package:

```
bluetooth.lib btdevice.lib btengdevman.lib<br>
btengdiscovery.lib btengconnman.lib btmanclient.lib<br>
btengsettings.lib sdpagent.lib sdpdatabase.lib<br>
```

Use cases

The API can be used to:

- get/set the Bluetooth device local name
- get/set the Bluetooth power state
- get/set the Bluetooth visibility mode
- add the device to the Bluetooth Registry
- delete the device from the Bluetooth Registry
- get all the devices from the Bluetooth Registry

Example code

The following code snippet can be used to retrieve the Bluetooth device local name:

```
CBTEngSettings* lBtSettings = CBTEngSettings::NewL();
CleanupStack::PushL(lBtSettings);
TBuf<32> lBTLocalName;
TInt lErrCode = lBtSettings->GetLocalName(lBTLocalName);
if ( lErrCode == KErrNone )
    {
    CEikonEnv::InfoWinL(_L("BT Name"),lBTLocalName);
    lBTLocalName.Zero();
    }
    else
    {
       TBuf<8> lErrCodeBuf;
    lErrCodeBuf.AppendNum(lErrCode);
    CEikonEnv::InfoWinL(_L("GetLocalBTName Error "), lErrCodeBuf);
    lErrCodeBuf.Zero();
    }
CleanupStack::PopAndDestroy(lBtSettings)
```

The following code snippet can be used to delete all Bluetooth devices from the registry:

(C) Copyright Nokia 2013. All rights reserved.

```
// Derive your class from M-Class MBTEngDevManObserver and implement
// all virtual functions
iBTEngDevMan = CBTEngDevMan::NewL(this);
// Search criteria find and delete all
TBTRegistrySearch lSearchCriteria;
lSearchCriteria.FindAll();
TInt lErrCode = iBTEngDevMan->DeleteDevices(lSearchCriteria);
if ( lErrCode == KErrNone )
        {
        CEikonEnv::InfoWinL(_L("Deletion"),_L("successful"));
        }
    else
        {
        TInt lErrCodeBuf;
        lErrCodeBuf.AppendNum(lErrCode );
        CEikonEnv::InfoWinL(_L("DeleteBTDevices Error"), lErrCodeBuf);
        }
```

Example project

The following example application needs the WriteDeviceData capability and the btengsettings.lib library. The following features have been implemented in this application:

- get/set Bluetooth device local name
- get/set Bluetooth power state
- get/set Bluetoth visibility mode

The application has been tested with the Nokia 6210 Navigator.

File:BTEngineExFor3rdEdFP2.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.