Archived:Getting list of paired 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

Getting a list of paired Bluetooth devices

Description

In order to retrieve a list of paired Bluetooth devices, design an Engine which derives from the "CActive" class.

Search for Bluetooth devices in the BT registry can be narrowed down with a `TBTRegistrySearch` instance.

The following code example uses the `TBTRegistrySearch::FindBonded()` criterion, which will return only Bluetooth devices that have already been paired with this device. Other criteria include searching by a specific device address, name, or Bluetooth device class. Search can be also be limited to return only trusted (authorized) devices.

Multiple search criteria can be used, in which case they are combined with the AND operator. `TBTRegistrySearch::FindAll()` will return all devices (unless other criteria are used).

```
void CBtEngine::GetPairedDevices( CBTDeviceArray& aDeviceArray )
    {
    RBTRegServ regServer;
    RBTRegistry registry;
    User::LeaveIfError( regServer.Connect() );
    CleanupClosePushL( regServer );
    User::LeaveIfError( registry.Open(regServer) );
    CleanupClosePushL( registry );
    TBTRegistrySearch searchPattern;
    searchPattern.FindBonded();
    registry.CreateView( searchPattern, iStatus );
    iState = ECreateView;
    SetActive();
    iSchedulerWait->Start(); // of type CActiveSchedulerWait*
    TInt retVal = iStatus.Int();
    if ( retVal > 0 )
        {
        CBTRegistryResponse* response =
            CBTRegistryResponse::NewL( registry );
        CleanupStack::PushL( response );
        response->Start( iStatus );
        iState = EGetResponse;
        SetActive();
        iSchedulerWait->Start();
        retVal = iStatus.Int();
        if ( retVal == KErrNone )
            {
            RBTDeviceArray results = response->Results();
            aDeviceArray.ResetAndDestroy();
            // RBTEngUtil::SortBTDevicesByNameL( results );

            TInt count = results.Count();
            for ( TInt i = 0; i < count; i++ )
                {
                aDeviceArray.AppendL( results[i]->CopyL() );
                }
            }
        CleanupStack::PopAndDestroy( response );
        }
    CleanupStack::PopAndDestroy( &registry );
    CleanupStack::PopAndDestroy( &regServer );
    }
```

`CActive::RunL()` implementation of `CBtEngine` class:

```
void CBtEngine::RunL()
    {
    switch( iState )
        {
        case ENone:
            break;
        case ECreateView:
            iSchedulerWait->AsyncStop();
            break;
        case EGetResponse:
            iSchedulerWait->AsyncStop();
            break;
        default:
            break;
        }
```


(C) Copyright Nokia 2013. All rights reserved.

```
}
```


(C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.