

## Get\_List\_of\_Active\_Bluetooth\_Connections

Symbian OS v7.0 have limited support for getting the connection information. There is a socket IOCTL command to get all the device addresses for connected ACL links (see KLMGetACLLinkArray in the Symbian Developer Library). Note that you can get the count of links first using KLMGetACLLinkCount. There is no way to get a notification when the set of links change.

The example code below shows how you can query for the list of addresses:

### Headers Required:

```
#include <e32std.h> //TPckgBuf
#include <bttypes.h> //TBTDevAddr
#include <es_sock.h> //RSocket
```

### Library needed:

```
LIBRARY euser.lib //TPckgBuf
LIBRARY bluetooth.lib //TBTDevAddr
```

### Source:

```
//Define buffer for the number of connected devices
TPckgBuf<TInt> theDevNo;

// Get the number of ACL links
TInt err = iSocket.GetOpt(KLMGetACLLinkCount, KSolBtLM, theDevNo);
User::LeaveIfError(err);

// Number of ACL connections
TInt& noLinks = theDevNo();
if(noLinks == 0)
{
return KErrNone; // If no links, no problem
}

// Allocate memory for the response of TBTDevAddr array
TBTDevAddr* tmpAddrArray = new(ELeave) TBTDevAddr[noLinks];
CleanupArrayDeletePushL(tmpAddrArray);

TInt theArraySize = noLinks*sizeof(TBTDevAddr);
TPtr8 ptr((TUint8*)tmpAddrArray, theArraySize, theArraySize);

// Get an array of device addresses for the ACL links, into a Link Array
err = iSocket.GetOpt(KLMGetACLLinkArray, KSolBtLM, ptr);
User::LeaveIfError(err);

// Make sure we iterate through the addresses that we did get back as
// opposed to the maximum which we could
noLinks = ptr.Length()/sizeof(TBTDevAddr);
CleanupStack::PopAndDestroy(); //tmpAddrArray
```

Symbian OS v8 provides improved support for this functionality. The "publish & subscribe" (P&S) mechanism can be used to register for notification when the number of physical links changes. The relevant P&S key is KPropertyKeyBluetoothLocalDeviceAddress, which is described in the API reference documentation as "The key to observe the the number of Bluetooth physical links attached to the local device".

## Get\_List\_of\_Active\_Bluetooth\_Connections

Information on the Publish & Subscribe framework is available in the Symbian OS v8 SDKs on Symbian DevNet - see "Series 60 2nd Edition with Feature Pack 2 (Symbian OS v8.0)" » Developer Library » Symbian OS Guide » C++ API guide » Base » Publish & Subscribe » Publish and Subscribe Overview

Code can then use the

```
CBluetoothPhysicalLinks::Enumerate()
```

method to get the addresses of the connected devices.

API reference for the method is:

```
//Enumerate members of the piconet
EXPORT_C TInt CBluetoothPhysicalLinks::Enumerate(RBTDevAddrArray& aBTDevAddrArray, TInt aMaxNumber)

//aBTDevAddrArray - Bluetooth device address array to be filled with bluetooth addresses of members
//aMaxNumber - Upper limit on number of members to be returned.
//Returns - Error code
```