WLAN Settings UI API

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

Purpose

WLAN settings UI API provides necessary functionality for launching a WLAN settings view.

It also gives the Comms Db column names to set the security settings of WLAN access point.

Use cases

To create a WLAN access point.

Example code

For example lets see how to create a WLAN access point with WEP security settings:

Headers required

```
#include <commdb.h>
#include <apselect.h>
#include <aplistitem.h>
#include <apdatahandler.h>
#include <apaccesspointitem.h>
#include <wlancdbcols.h>   // for security settings
```

Libraries required

```
LIBRARY  commdb.lib apengine.lib
```

Source Code

```
CApAccessPointItem *wlan = CApAccessPointItem::NewLC();
wlan->SetNamesL(_L("NewAP"));
wlan->SetBearerTypeL(EApBearerTypeWLAN);
wlan->WriteTextL(EApWlanNetworkName, _L("WlanAP"));
wlan->WriteUint(EApWlanSecurityMode,2); // security mode 2 refers to WEP.
wlan->WriteUint(EApWlanNetworkMode,1);
// network mode refers to whether its AdHoc or Infrastructure mode.

// Store it into the CommsDb
CCommsDatabase *commDb = CCommsDatabase::NewL();
CleanupStack::PushL(commDb);
CApDataHandler *handler = CApDataHandler::NewLC(*commDb);
TInt err = commDb ->BeginTransaction();
TUint32 newApId = handler->CreateFromDataL(*wlan);
CleanupStack::PopAndDestroy(3); // handler, commDb, wlan

//Setting the security settings:

enum TWEPKeyFormat
    {
    EAscii,                 // Ascii format
    EHexadecimal            // Hex format
    };

    enum TWEPKeyInUse
    {
    EKeyNumber1,            // Key number 1
    EKeyNumber2,            // Key number 2
    EKeyNumber3,            // Key number 3
    EKeyNumber4             // Key number 4
    };

enum TWEPAuthentication
    {
    EAuthOpen,              // Open authentication
    EAuthShared             // Shared authentication
    };

handler->AccessPointDataL(newApId,*wlanNew1);
//wlanNew1 is an object of CApAccessPointItem
wlanNew1->ReadUint( EApIapServiceId, newalanid );
// we need to get the IAP id to manupulate the security settings.

iKeyData.Copy(_L("626ABB616A"));
//Should be the same value to which the WLAN router is configured

iKeyFormat=EAscii;           // TWEPKeyFormat
iAuthentication=EAuthShared; // TWEPAuthentication
iKeyInUse=EKeyNumber3;       // TWEPKeyInUse

wLanServiceTable = commDb->OpenViewMatchingUintLC( TPtrC( WLAN_SERVICE ),
```


(C) Copyright Nokia 2013. All rights reserved.

```
TPtrC( WLAN_SERVICE_ID ), newalanid );  // Give the corresponding IAP ID

errorCode = wLanServiceTable->GotoFirstRecord();

TBool ival =ETrue;
 if ( errorCode == KErrNone )
        {
        wLanServiceTable->UpdateRecord();

        }
    else
        {
        TUint32 dummyUid( KUidNone );
        User::LeaveIfError( wLanServiceTable->InsertRecord( dummyUid ) );
        // Save link to LAN service
 wLanServiceTable->WriteUintL( TPtrC( WLAN_SERVICE_ID ), newApId);
        }

// Save index of key in use (1,2,3 or 4)
TRAP(error,wLanServiceTable->WriteUintL( TPtrC( WLAN_WEP_INDEX ),
( TUint32& ) iKeyInUse ));

// Save the authentication type (shared or Open)
TRAP(error,wLanServiceTable->WriteUintL( TPtrC( WLAN_AUTHENTICATION_MODE ),
( TUint32& )iAuthentication ));

// save the correct keydata corresponding to the router configuration
TRAP(error,wLanServiceTable->WriteTextL( TPtrC( WLAN_WEP_KEY3 ), iKeyData ));

// Save the format of the key (Hexadecimal or Ascii)
wLanServiceTable->WriteUintL( TPtrC( LAN_WEP_KEY3_FORMAT ),
( TUint32& ) iKeyFormat );

wLanServiceTable->PutRecordChanges();

err = commDb->CommitTransaction();
// End a transaction. Call after `InsertRecord()` or `UpdateRecord()`.
```

NOTE: The Key In Use, the Key format and the Key data should be the same as the WLAN router or else the connection will not be established.

Example project

File:WLanEx.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.