How to show WLAN IAP only when the device is offline

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*}}.

The article is believed to be still valid for the original topic scope.

Overview

When the device is in offline mode, the application must show only the WLAN Internet Access Point (IAP) in the IAP selection dialog. This functionality can be implemented with the `CApSettingsHandler` class.

This snippet can be self-signed.

MMP file

The following capabilities and libraries are required:

CAPABILITY NetworkServices

LIBRARY apsettingshandlerui.lib // for CApSettingsHandler
LIBRARY centralrepository.lib // for Offline/Online check
LIBRARY apengine.lib // for getting Easy WLAN

Header file

```
#include <apsettingshandlerui.h>    // EApBearerTypeWLAN
#include <centralrepository.h>      // CRepository
#include <ProfileEngineSDKCRKeys.h> // CRepository
#include <activeapdb.h>             // CActiveApDb

// Offline profileid
const TInt KOffline = 5;

// Class member variables
TInt iPrevProfileId;
TBool iConnectionSetupDone;
RSocketServ iSocketServ;
RConnection iConnection;
RHTTPSession iSession;
```

Source file

In the connection setup method of the application (`SetupConnectionL()`), check the device profile by using `CRepository`. If the profile is Offline (5), show only WLANs of type `EApBearerTypeWLAN` in the IAP selection dialog.

If the profile is changed after the first IAP selection, the IAP checked again.

```
void CYourClass::SetupConnectionL()
    {
    TUint32 selectedIap = 0;
    TInt bearerFilter = EApBearerTypeAllBearers;
    TInt currentProfileId;

    // Create CRepository
    CRepository repository = CRepository::NewL(KCRUidProfileEngine);

    // Check whether device is offline or online
    repository->Get(KProEngActiveProfile, currentProfileId);
    delete repository;

    // Close the connection only if
    // a) this is not the first time and
    // b) the profile has changed and
    // c) either the previous or the current profile are Offline
    if (iPrevProfileId != -1 && iPrevProfileId != currentProfileId &&
       (iPrevProfileId == KOffline || currentProfileId == KOffline))
        {
        // Close and uninitialize
        iConnectionSetupDone = EFalse;
        iSession.Close();
        iConnection.Close();
        iSocketServ.Close();
        }

    // Save the current profile ID
    iPrevProfileId = currentProfileId;
```


(C) Copyright Nokia 2013. All rights reserved.

```
    if (iConnectionSetupDone)
        {
        // Connection setup is done
        return;
        }

    // Open RHTTPSession with default protocol ("HTTP/TCP")
    iSession.OpenL();

    // In offline, only WLAN connections are available
    if (currentProfileId == KOffline)
        {
        bearerFilter = EApBearerTypeWLAN;
        }

    // Create the IAP selection dialog
    CActiveApDb* aDb = CActiveApDb::NewL();
    CleanupStack::PushL(aDb);
    CApSettingsHandler* settings = CApSettingsHandler::NewLC(
            *aDb,
            ETrue,
            EApSettingsSelListIsPopUp,
            EApSettingsSelMenuSelectNormal,
            KEApIspTypeAll,
            bearerFilter,
            KEApSortNameAscending,
            0,
            EVpnFilterBoth,
            ETrue);

    // Show the dialog
    TInt iapRet = settings->RunSettingsL(0, selectedIap);
    CleanupStack::PopAndDestroy(settings);
    CleanupStack::PopAndDestroy(aDb);
    if (iapRet != KApUiEventSelected)
        {
        // Exit: no selection
        User::Leave(KErrNotReady);
        }
    else
        {
        // IAP Selected
        // Open socket server and start the connection
        User::LeaveIfError(iSocketServ.Connect());
        User::LeaveIfError(iConnection.Open(iSocketServ));

        // Now you have the IAP ID. Use it to connect to the connection.
        TCommDbConnPref connectPref;
        // Setup preferences
        connectPref.SetDialogPreference(ECommDbDialogPrefDoNotPrompt);
        // Set the CommDb ID of the IAP to use for this connection
        connectPref.SetIapId(selectedIap);
        // Start connection
        User::LeaveIfError(iConnection.Start(connectPref));

        // Set the sessions connection info...
        RStringPool strPool = iSession.StringPool();
        RHTTPConnectionInfo connInfo = iSession.ConnectionInfo();
        // ...to use the socket server and connection
        connInfo.SetPropertyL(
            strPool.StringF(HTTP::EHttpSocketServ, RHTTPSession::GetTable()),
            THTTPHdrVal(iSocketServ.Handle())
        );
        // ...to use the connection
        connInfo.SetPropertyL(
            strPool.StringF(HTTP::EHttpSocketConnection,
                RHTTPSession::GetTable()),
            THTTPHdrVal(REINTERPRET_CAST(TInt, &(iConnection)))
        );

        iConnectionSetupDone = ETrue;
        }
    }
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.