

Symbian Zugangspunkte auslesen

Geschrieben von: Philipp

MONDAY, 10 MARCH 2008 03:21 - Zuletzt aktualisiert THURSDAY, 13 MARCH 2008 00:43

Gestet auf Symbian 6 - 8 Series 60, Series 80 und UIQ

Im folgenden zeige ich, wie auf einem Symbian Gerät die Zugangspunkte (Einwahlpunkte für den Zugang zum Internet) ausgelesen werden und diese dann zur Weiterverarbeitungen in einem CArrayFixFlat alphabetisch sortiert zurückgegeben werden können.

Die Zugangspunkte werden unter Symbian intern in CommDatabase gespeichert. Es handelt sich dabei um ein proprietäres Symbian SQL System, auf welches man theoretisch auch mit RDBNamedDatabase zugreifen könnte. Symbian stellt uns aber speziell für die CommDatabase die Klasse CCommsDatabase zur Verfügung.

Ersteinmal includieren wir das benötigte header-file und erstellen uns ein struct, welches die relevanten Zugangspunktendaten für die Sortierung beinhaltet: #include <commdb.h>

```
struct TAccesspointInfo
{
    TUint32    APID;
    TBuf<40>    APName;
};
```

Wir erstellen uns ein CCommsDatabase Object für alle Internet-Access-Points (EDatabaseTypeIAP) //

connect the CommsDatabase

```
CCommsDatabase* dataBase = CCommsDatabase::NewL(EDatabaseTypeIAP);
CleanupStack::PushL(dataBase);
```

Nun legen wir die Filter fest, um die Art der Zugangspunkte, welche wir zurück bekommen wollen, zu definieren: // define the filter

```
TInt filter = ECommDbBearerGPRS | ECommDbBearerLAN; ACHTUNG: der Filtertyp ECommDbBearerLAN
```

ist erst seit Symbian 7 verfügbar. Es werden mit diesen zusätzlich WLAN-Zugangspunkte gewählt. ECommDbBearerGPRS steht für alle Zugangspunkte die über GRPS oder auch UMTS eine Verbindung herstellen.

Als nächstes erstellen wir uns eine view vom Typ CCommsDbTableView, welche wir auch gleich angewandt auf die Datenbank ausführen um somit unsere Zugangspunkte zu erhalten // the view

```
CCommsDbTableView *view =
dataBase->OpenIAPTableViewMatchingBearerSetLC(filter,ECommDbConnectionDirectionOutgoing);
```

Was wir jetzt noch brauchen, ist das im Endeffekt zurück zugebene Array: // fill the array for sorting

```
CArrayFixFlat<TAccesspointInfo>* sortedArr = new (ELeave)
CArrayFixFlat<TAccesspointInfo>(1);
CleanupStack::PushL(sortedArr);
```

Nun die eigentlichen Daten aus der View auslesen und dem Array hinzufügen: // add the

Symbian Zugangspunkte auslesen

Geschrieben von: Philipp

MONDAY, 10 MARCH 2008 03:21 - Zuletzt aktualisiert THURSDAY, 13 MARCH 2008 00:43

accesspoints to the array

```
TInt apIndex = view->GotoFirstRecord();
```

```
while (apIndex == KErrNone)
```

```
{
```

```
    TAccesspointInfo apInfo;
```

```
    // get the name
```

```
    view->ReadTextL(TPtrC(COMMDB_NAME), apInfo.APName);
```

```
    // get the ID
```

```
    TUint32 ID = 0;
```

```
    view->ReadUIntL(TPtrC(COMMDB_ID), apInfo.APID);
```

```
    // add to array
```

```
    sortedArr->AppendL(apInfo);
```

```
    apIndex = view->GotoNextRecord();
```

```
}
```

Jetzt sind wir an den Punkt angekommen, an welchen wir das Array alphabetisch sortieren können: // sort the array

```
TKeyArrayFix sortKey(_FOFF(TAccesspointInfo, APName), ECmpCollated);
```

```
sortedArr->Sort(sortKey);
```

Als letztes nur noch aufräumen und das Ergebniss zurück geben: // cleanup

```
CleanupStack::Pop(); // sortedArr
```

```
CleanupStack::PopAndDestroy(2); // view, dataBase
```

```
return sortedArr;
```

Das Ganze könnte dann in einer Funktion wie folgt aussehen: #include <commdb.h>

```
// needed for AP-Stuff
```

```
struct TAccesspointInfo
```

```
{
```

```
    TUint32    APID;
```

```
    TBuf<40>   APName;
```

```
};
```

```
/**
```

```
* returns an array with accesspoints and the ID of every accesspoint
```

```
**/
```

```
CArrayFixFlat<TAccesspointInfo>* GetAccesspointsL()
```

```
{
```

```
    // connect the CommsDatabase
```

```
    CCommsDatabase* dataBase = CCommsDatabase::NewL(EDatabaseTypeIAP);
```

```
    CleanupStack::PushL(dataBase);
```

```
    // define the filter
```

Symbian Zugangspunkte auslesen

Geschrieben von: Philipp

MONDAY, 10 MARCH 2008 03:21 - Zuletzt aktualisiert THURSDAY, 13 MARCH 2008 00:43

```
TInt filter = ECommDbBearerGPRS | ECommDbBearerLAN;

// the view
CCommsDbTableView *view =
dataBase->OpenIAPTableViewMatchingBearerSetLC(filter,ECommDbConnectionDirectionOutgoing);

// fill the array for sorting
CArrayFixFlat<TAccesspointInfo>* sortedArr = new (ELeave)
CArrayFixFlat<TAccesspointInfo>(1);
CleanupStack::PushL(sortedArr);

// add the accesspoints to the array
TInt apIndex = view->GotoFirstRecord();

while (apIndex == KErrNone)
{
    TAccesspointInfo apInfo;

    // get the name
    view->ReadTextL(TPtrC(COMMDB_NAME), apInfo.APName);

    // get the ID
    TUint32 ID = 0;
    view->ReadUIntL(TPtrC(COMMDB_ID), apInfo.APID);

    // add to array
    sortedArr->AppendL(apInfo);
    apIndex = view->GotoNextRecord();
}

// sort the array
TKeyArrayFix sortKey(_FOFF(TAccesspointInfo, APName), ECmpCollated);
sortedArr->Sort(sortKey);

// cleanup
CleanupStack::Pop(); // sortedArr
CleanupStack::PopAndDestroy(2); // view, dataBase

return sortedArr;
}
```