



Contents

- [1 Creating GPRS access point](#)
- [2 Mandatory tables](#)
- [3 Example project](#)
- [4 Related Links](#)

Creating GPRS access point

Headers Required:

```
#include <commdb.h> //CCommsDatabase, CCommsDbTableView
```

Library required:

```
LIBRARY commdb.lib //CCommsDatabase, CCommsDbTableView
```

This function creates a new IAP record and insert it into the Comms database:

```
TInt CIAPConnect_v30Engine::CreateNewIapL(TDesC& aIapName)
{
    CCommsDatabase* commsDb;
    commsDb = CCommsDatabase::NewL();
    CleanupStack::PushL(commsDb);
    TBufC<32> commsdb_name_val(aIapName);
    TUint32 network_id;
    commsDb->BeginTransaction();

    /* Step 1 */
    // creating a new network record
    CCommsDbTableView* network = commsDb->OpenTableLC(TPtrC(NETWORK));
    network->InsertRecord(network_id);
    network->WriteTextL(TPtrC(COMMDB_NAME), commsdb_name_val);
    network->PutRecordChanges();
    CleanupStack::PopAndDestroy(network);

    /* Step 2 */
    // creating a new outgoing gprs record
    TUint32 og_gprs_id;

    // See implementations in the CreateNewOgGprsL function
    TInt err= CreateNewOgGprsL(*commsDb, commsdb_name_val, og_gprs_id);

    /* Step 3 */
    // creating a new wap accesspoint record
    TUint32 ap_id;

    // See implementations in the CreateNewAccessPointL function
    err = CreateNewAccessPointL(*commsDb, commsdb_name_val, ap_id);

    /* Step 4 */
    // creating a new IAP record
    CCommsDbTableView* iap = commsDb->OpenTableLC(TPtrC(IAP));
    TUint32 iap_id;
```

Creating_a_GPRS_access_point_â _Mandatory_Tables

```
err = iap->InsertRecord(iap_id);

iap->WriteTextL(TPtrC(COMMDB_NAME), commsdb_name_val);
iap->WriteUIntL(TPtrC(IAP_SERVICE), og_gprs_id);
iap->WriteTextL(TPtrC(IAP_SERVICE_TYPE), TPtrC(OUTGOING_GPRS));
iap->WriteTextL(TPtrC(IAP_BEARER_TYPE), TPtrC(MODEM_BEARER));

// In real application, don't hard-code. Should check from the modem bearer
iap->WriteUIntL( TPtrC(IAP_BEARER), 2);

iap->WriteUIntL( TPtrC(IAP_NETWORK), network_id);
iap->WriteUIntL( TPtrC(IAP_NETWORK_WEIGHTING), 0 );
iap->WriteUIntL( TPtrC(IAP_LOCATION), 2);
iap->PutRecordChanges();
CleanupStack::PopAndDestroy(iap);

/* Step 5 */
// creating a new wap bearer
CCommsDbTableView* wap_bearer = commsDb->OpenTableLC(TPtrC(WAP_IP_BEARER));
TUInt32 wb_id;
wap_bearer->InsertRecord(wb_id);
wap_bearer->WriteUIntL(TPtrC(WAP_ACCESS_POINT_ID), ap_id);
_LIT(wap_gw_address, "0.0.0.0");
wap_bearer->WriteTextL(TPtrC(WAP_GATEWAY_ADDRESS), wap_gw_address);

wap_bearer->WriteUIntL( TPtrC(WAP_IAP), iap_id);
wap_bearer->WriteUIntL( TPtrC(WAP_WSP_OPTION), EWapWspOptionConnectionOriented);

wap_bearer->WriteBoolL( TPtrC(WAP_SECURITY), EFalse);
wap_bearer->WriteUIntL( TPtrC(WAP_PROXY_PORT), 0 );

wap_bearer->PutRecordChanges();
CleanupStack::PopAndDestroy(wap_bearer);
// Finish

err= commsDb->CommitTransaction();
CleanupStack::PopAndDestroy(); // commsDb

return KErrNone;
}
```

The following function creates a new outgoing GPRS record:

```
TInt CIAPConnect_v30Engine::CreateNewOgGprsL(CCommsDatabase& aCommsDb,
                                             TDesC& aCommsdb_name_val, TUInt32& aOgGprsId)
{
    CCommsDbTableView* commsOgGprs = aCommsDb.OpenTableLC(TPtrC(OUTGOING_GPRS));
    TInt err = commsOgGprs->InsertRecord(aOgGprsId);

    commsOgGprs->WriteTextL(TPtrC(COMMDB_NAME), aCommsdb_name_val);

    _LIT(apn_val, "airtelgprs.com");
    commsOgGprs->WriteLongTextL(TPtrC(GPRS_APN), apn_val);

    commsOgGprs->WriteUIntL( TPtrC(GPRS_PDP_TYPE), 0);

    commsOgGprs->WriteUIntL( TPtrC(GPRS_REQ_PRECEDENCE), 0);

    commsOgGprs->WriteUIntL( TPtrC(GPRS_REQ_DELAY), 0);
}
```

Creating_a_GPRS_access_point_â_Mandatory_Tables

```
commsOgGprs->WriteUIntL( TPtrC(GPRS_REQ_RELIABILITY), 0);
commsOgGprs->WriteUIntL( TPtrC(GPRS_REQ_PEAK_THROUGHPUT), 0);
commsOgGprs->WriteUIntL( TPtrC(GPRS_REQ_MEAN_THROUGHPUT), 0);
commsOgGprs->WriteUIntL( TPtrC(GPRS_MIN_PRECEDENCE), 0);
commsOgGprs->WriteUIntL( TPtrC(GPRS_MIN_DELAY), 0);
commsOgGprs->WriteUIntL( TPtrC(GPRS_MIN_RELIABILITY), 0);
commsOgGprs->WriteUIntL( TPtrC(GPRS_MIN_PEAK_THROUGHPUT), 0);
commsOgGprs->WriteUIntL( TPtrC(GPRS_MIN_MEAN_THROUGHPUT), 0);
commsOgGprs->WriteBoolL( TPtrC(GPRS_DATA_COMPRESSION), EFalse);
commsOgGprs->WriteBoolL( TPtrC(GPRS_HEADER_COMPRESSION), EFalse);
commsOgGprs->WriteBoolL( TPtrC(GPRS_ANONYMOUS_ACCESS), EFalse);
commsOgGprs->WriteTextL( TPtrC(GPRS_IF_NETWORKS), _L("ip"));
commsOgGprs->WriteBoolL( TPtrC(GPRS_IF_PROMPT_FOR_AUTH), EFalse);
commsOgGprs->WriteUIntL( TPtrC(GPRS_IF_AUTH_RETRIES), 0);
commsOgGprs->WriteTextL( TPtrC(GPRS_IP_GATEWAY), _L("0.0.0.0"));
commsOgGprs->WriteBoolL( TPtrC(GPRS_IP_DNS_ADDR_FROM_SERVER), EFalse);
commsOgGprs->WriteTextL( TPtrC(GPRS_IP_NAME_SERVER1), _L("0.0.0.0"));
commsOgGprs->WriteTextL( TPtrC(GPRS_IP_NAME_SERVER2), _L("0.0.0.0"));
commsOgGprs->WriteBoolL( TPtrC(GPRS_ENABLE_LCP_EXTENSIONS), EFalse);
commsOgGprs->WriteBoolL( TPtrC(GPRS_DISABLE_PLAIN_TEXT_AUTH), ETrue);
commsOgGprs->WriteBoolL( TPtrC(GPRS_IP_ADDR_FROM_SERVER), ETrue);
commsOgGprs->WriteUIntL( TPtrC(GPRS_AP_TYPE), 2);
commsOgGprs->WriteUIntL( TPtrC(GPRS_QOS_WARNING_TIMEOUT), -1);
commsOgGprs->PutRecordChanges();
CleanupStack::PopAndDestroy(commsOgGprs);
return KErrNone;
}
```

This function creates a new wap access point:

```
TInt CIAPConnect_v30Engine::CreateNewAccessPointL(CCommsDatabase& aCommsDb,
                                                  TDesC& aCommsdb_name_val, TUint32& aApId)
{
    CCommsDbTableView* commsAp = aCommsDb.OpenTableLC(TPtrC(WAP_ACCESS_POINT));
    commsAp->InsertRecord(aApId);
}
```

Creating_a_GPRS_access_point_â _Mandatory_Tables

```
commsAp->WriteTextL(TPtrC(COMMDB_NAME), aCommsdb_name_val);  
commsAp->WriteTextL(TPtrC(WAP_CURRENT_BEARER), TPtrC(WAP_IP_BEARER));  
commsAp->PutRecordChanges();  
CleanupStack::PopAndDestroy(commsAp);  
return KErrNone;  
}
```

Mandatory tables

The [CommDb](#) tables required to set up a connection are described in

[Symbian Developer Knowledgebase](#)

[FAQ-1173](#).

Example project

[Example project to create GPRS access point](#)

Related Links

[CommDb](#)

[Reading internet access points from the device](#)

[Create CSD IAP using CommDB API](#)

[How to create access point](#)

[TSS000418 - Creating, deleting, editing, and listing WLAN access points](#)