

NOTE

The source of this information is Symbian's [FAQ-0970](#) but a key information from that article is missing: the sample code provided below is intended for "**Symbian OS 7.0 (and later) Binary Access Kit / DevKit licensees**".

What that basically means is that the APIs used are not public and can only be accessed through Symbian's partnership programs. It also means that a S60 developer using the public SDK cannot effectively apply this solution for accessing the own number from their code. Since the API is owned by Symbian it is also not accessible through Forum Nokia's S60 API Partnering program.

Bottom line, the solution described in this article is useless for most of the developers that are using this wiki.

Please take this article as an example and try to make sure that the articles you are contributing to are:

- based on publicly available official information
- using only public APIs

Forum Nokia reserves the right to suspend or remove articles that contain confidential information or make use of internal APIs.

How to get my own phone number (eventually)

```
void RMobileONStore::Read(TRequestStatus& aReqStatus, TDes8& aEntry)
```

This API is being used to get the own phone number:

```
// Requirement: etel server, tsy, have been opened and initialized
```

```
TBool isSupported = EFalse;
User::LeaveIfError(telServer.IsSupportedByModule(tsyName, KETelFuncMobileOwnNumberStore, isSupported));

if (isSupported != EFalse)
{
    console->Printf(_L("Own number store is supported.\n"));
}
else
{
    User::Leave(KErrNotSupported);
}

TInt numPhones=0;
telServer.EnumeratePhones(numPhones);
```

How_to_get_my_own_phone_number

```
for (TInt i=0; i<numPhones; i++)
{
    TRequestStatus status;
    RTelServer::TPhoneInfo info;
    telServer.GetPhoneInfo(i, info);

    RMobilePhone phone;
    User::LeaveIfError(phone.Open(telServer, info.iName));
    CleanupClosePushL(phone);

    RMobileONStore ownNumberStore;
    User::LeaveIfError(ownNumberStore.Open(phone));
    CleanupClosePushL(ownNumberStore);

    // Check if there is own number entry stored.
    RMobileONStore::TMobileONStoreInfoV1 ownStoreInfo;
    RMobileONStore::TMobileONStoreInfoV1Pckg ownStoreInfoPckg(ownStoreInfo);

    ownNumberStore.GetInfo(status, ownStoreInfoPckg);
    User::WaitForRequest(status);

    if(status.Int() == KErrNone)
    {
        CheckCapabilities(ownStoreInfo.iCaps);

        if(ownStoreInfo.iUsedEntries < 1)
        {
            console->Printf(_L("No entry found.\n"));
            User::Leave(KErrNotFound);
        }
        else
        {
            User::Leave(status.Int());
        }

        // Read the own number entry
        status = KRequestPending;
        RMobileONStore::TMobileONEntryV1 ownNumberEntry;
        ownNumberEntry.iIndex = 1;
        RMobileONStore::TMobileONEntryV1Pckg ownNumberEntryPckg(ownNumberEntry);

        ownNumberStore.Read(status, ownNumberEntryPckg);
        User::WaitForRequest(status);

        if (status.Int() == KErrNone)
        {
            console->Printf(_L("\nOwn number = %S"), &ownNumberEntry.iNumber.iTelNumber);
        }
        else
        {
            console->Printf(_L("\nNO Own number: error %d"), status);
        }

        CleanupStack::PopAndDestroy(2); // i-th phone & ownNumberStore;
    }
}
```

How_to_get_my_own_phone_number

Alternatively, you may be able to use a helper class `CRetrieveMobilePhoneONList` if the target device supports `KCapsWholeStore` capability. The following code snippet illustrates an example usage.

```
#include <mmretrieve.h>

// Precondition server, phone module and phone have been opened

RMobileONStore ownNumberStore;
User::LeaveIfError(store.Open(phone));
CleanupClosePushL(ownNumberStore);

RMobilePhoneBookStore::TMobilePhoneBookInfoV1 storeInfo;
RMobilePhoneBookStore::TMobilePhoneBookInfoV1Pckg storeInfoPckg(storeInfo);
store.GetInfo(status, storeInfoPckg);
User::WaitForRequest(status);

if(status == KErrNone)
{
if(storeInfo.iCaps & RMobilePhoneStore::KCapsWholeStore)
{
console->Printf(_L("KCapsWholeStore supported"));
}
else if (storeInfo.iCaps & RMobilePhoneStore::KCapsIndividualEntry )
{
console->Printf(_L("KCapsIndividualEntry supported"));
}
}

CRetrieveMobilePhoneONList* retrieve;
retrieve=CRetrieveMobilePhoneONList::NewL(ownNumberStore);

retrieve->Start(status);
User::WaitForRequest(status);

CMobilePhoneONList* onList=NULL;
TInt error=KErrNone;
TRAP(error, onList=retrieve->RetrieveListL());

TInt numOfEntry = 0;
if (error == KErrNone)
{
numOfEntry = onList->Enumerate();
}

RMobileONStore::TMobileONEntryV1 entry;
for (TInt index=0; index < numOfEntry ; index++)
{
error = KErrNone;
TRAP(error, entry = onList->GetEntryL(index))
if (error == KErrNone)
{
console->Printf(_L("Own Number %S\n"), entry.iNumber.iTelNumber);
}
else
{
console->Printf(_L("Read error!!"));
}
}

//Close the phone and the mobileStore and remove them from the cleanup stack
CleanupStack::PopAndDestroy(2);
```

How_to_get_my_own_phone_number

--