



Purpose

A mail entry has different components like "From", "Subject", "Body", "Attachments", etc, The purpose of this page is to retrieve these values.

Header Files

```
#include <msvapi.h> //CMsvSession, CMsvEntry
#include <cemailaccounts.h> //CEmailAccounts
#include <miutmsg.h> //CImEmailMessage
#include <miuthdr.h> //CImHeader
```

Libraries

```
LIBRARY msgs.lib
LIBRARY imcm.lib
```

Capabilities

```
Capability ReadUserData
```

Code Snippets

The following code is used to create a session synchronously to message server:

```
//Derive your class from "MMsvSessionObserver" and give implementation to the
//"HandleSessionEventL(TMsvSessionEvent, TAny*, TAny*, TAny*);" method.

iSession = CMsvSession::OpenSyncL(*this);
```

The following code is used to construct an Email accounts object and gets a list of POP account IDs.:

```
CEmailAccounts* mailAccount = CEmailAccounts::NewLC();
RArray<TPopAccount> accounts(10);

mailAccount->GetPopAccountsL(accounts);

TInt countAcct = accounts.Count();
TBuf<100> countAcctBuf;
countAcctBuf.AppendNum(countAcct);
CEikonEnv::InfoWinL(_L("Number of Accounts"), countAcctBuf);
```

Traverse through the available accounts and retrieve the entries in each account as done below.

The following code is used to retrieve the TMsvId of the particular account and retrieve its entries:

How_to_retrieve_e-mail_contents

```
//getting the TMsvId of the account
TPopAccount popAcnt = accounts.operator[](i);
TMsvId mailId = popAcnt.iPopService;

//getting the entry details in that account
CMsvEntry* mailEntry =CMsvEntry::NewL
(*iSession,mailId,TMsvSelectionOrdering::TMsvSelectionOrdering());
CleanupStack::PushL(mailEntry);

//getting the children entries
CMsvEntrySelection* entries = mailEntry->ChildrenL();
CleanupStack::PushL( entries );

//getting no. of mails
TInt mailCount= entries->Count();

TBuf<100> mailCountBuf;
mailCountBuf.AppendNum(mailCount);
CEikonEnv::InfoWinL(_L("Number of Mails"), mailCountBuf);
```

The following code is used to retrieve TMsvId of each mail and get the "From" and "Subject" from it from the store:

```
TMsvId entryID = entries->At(mailTraverese);
//mailTraverese is TInt starts from 0 to mailCount-1

CMsvEntry* entry= iSession->GetEntryL(entryID);
CleanupStack::PushL(entry);

CMsvStore *store = entry->ReadStoreL();
CleanupStack::PushL(store);

CImHeader* header = CImHeader::NewLC();
header->RestoreL(*store);

//subject buffer contains the "subject" of the mail.
TBuf<50> subject = header->Subject();

//header buffer contains the "header" of the mail.
TBuf<50> from = header->From();

CleanupStack::PopAndDestroy(2, store);//header, store
```

The following code is used to retrieve the mail Body

```
CRichText* richText;
CParaFormatLayer* iParaFormatLayer;
CCharFormatLayer* iCharFormatLayer;

paraFormatLayer=CParaFormatLayer::NewL();
charFormatLayer=CCharFormatLayer::NewL();

richText=CRichText::NewL(paraFormatLayer, charFormatLayer);
CleanupStack::PushL(richText);

CIMEmailMessage* mailMsg = CIMEmailMessage::NewLC(*entry);
mailMsg->GetBodyTextL(entryID, CIMEmailMessage::EThisMessageOnly,
 *richText , *iParaFormatLayer, *iCharFormatLayer);
```

How_to_retrieve_e-mail_contents

```
CleanupStack::PopAndDestroy(5, mailEntry); //mailMsg,richText,entry,entries,mailEntry
```

The following link is used to retrieve the attachment info. in the mail: [Retrieve Attachment Info.](#)

Example Project

[Retrieve Mail Contents](#)