

| | | | |
|---|----------------------|--------------------------|----------------|
| ID | | Creation date | March 22, 2008 |
| Platform | S60 3rd, S60 3rd FP1 | Tested on devices | N73, N95 8Gb |
| Category | Symbian C++ | Subcategory | Messaging |
| Keywords: Unread messages, unread emails | | | |

Overview

The original article is in Russian. You can find it [here](#).

You can use class **CUnreadMsgCalc** for getting the count of unread messages and for tracking appearance of new messages.

This class also separately counts the number of unread emails from all mailboxes, registered on the current device (only already downloaded emails checked).

In order to obtain the required information, this class asynchronously creates session for interaction with the Message Server. This session is initialized in `ConstructL()`.

Main methods:

- `NewMsgCount()` returns count of unread messages
- `NewEmailCount()` returns count of unread emails
- `IsCalcCompleted()` returns check result (calculation is completed or not)

Header File

```
#include <msvapi.h>    //MsvSessionObserver

class CUnreadMsgCalc: public CBase,
                    public MsvSessionObserver
{
public:
    static CUnreadMsgCalc* NewL(); // factory
    virtual ~CUnreadMsgCalc();

    inline TBool IsCalcCompleted() { return iCalcCompleted; } // if true - then NewXXXCount re

    inline TInt NewMsgCount()      { return iNewMsgCount; }
    inline TInt NewEmailCount()    { return iNewEmailPopCount + iNewEmailImapCount; }

private:
    void ConstructL();
    void CalcUnreadEntriesL( const TMsvid aMsvId, TInt& aCounter ); // read one entry

private: // from MsvSessionObserver
    void HandleSessionEventL( TMsvid aEvent, TAny* aArg1, TAny* aArg2, TAny* aArg3 );

private:
```

How_to_get_the_count_of_different_unread_messages

```
CMsvSession* iMsvSession;  
TBool iCalcCompleted;  
TInt iNewMsgCount, iNewEmailPopCount, iNewEmailImapCount;  
};
```

Source File

```
#include "msgCalc.h"  
  
#include <cemailaccounts.h> //CEmailAccounts  
#include <miutmsg.h> //CImEmailMessage  
#include <miuthdr.h> //CImHeader  
  
CUnreadMsgCalc* CUnreadMsgCalc :: NewL()  
{  
    CUnreadMsgCalc* self = new ( ELeave ) CUnreadMsgCalc;  
    CleanupStack :: PushL( self );  
    self->ConstructL();  
    CleanupStack :: Pop( self );  
    return self;  
}  
  
CUnreadMsgCalc :: ~CUnreadMsgCalc()  
{  
    delete iMsvSession;  
    iMsvSession = NULL;  
}  
  
void CUnreadMsgCalc :: ConstructL()  
{  
    iMsvSession = CMsvSession :: OpenAsyncL( *this );  
}  
  
void CUnreadMsgCalc :: CalcUnreadEntriesL( const TMsvId aMsvId, TInt& aCounter )  
{  
    aCounter = 0;  
    CMsvEntry* entry = CMsvEntry :: NewL( *iMsvSession, aMsvId, TMsvSelectionOrdering() );  
    CleanupStack::PushL( entry );  
    CMsvEntrySelection* entries = entry->ChildrenL();  
    CleanupStack::PushL( entries );  
    for( TInt i = 0; i < entries->Count(); i++ )  
    {  
        entry->SetEntryL( entries->At( i ) );  
        TMsvEntry msvEntry( entry->Entry() );  
        if( msvEntry.Unread() )  
            aCounter++;  
    }  
    CleanupStack::PopAndDestroy( 2 ); // entries entry  
}  
  
void CUnreadMsgCalc :: HandleSessionEventL( TMsvSessionEvent aEvent, TAny* aArg1, TAny* aArg2, TA  
{  
    switch( aEvent )  
    {  
        case EMsvServerReady:  
        case EMsvEntriesCreated:  
        case EMsvEntriesChanged:  
        case EMsvEntriesDeleted:
```

How_to_get_the_count_of_different_unread_messages

```
case EMsvEntriesMoved:
{
    iCalcCompleted = EFalse;
    CalcUnreadEntriesL      ( KMsvGlobalInBoxIndexEntryId, iNewMsgCount ); // reading global

    CEmailAccounts* mailAccount = CEmailAccounts::NewLC(); // get email accounts list

    iNewEmailPopCount = 0;
    RArray< TPopAccount > popAccounts( 5 );
    mailAccount->GetPopAccountsL( popAccounts );
    CleanupClosePushL( popAccounts );
    for( TInt i = 0; i < popAccounts.Count(); i++ ) // reading all email inboxes (POP)
    {
        TInt iCounter = 0;
        CalcUnreadEntriesL( popAccounts[i].iPopService, iCounter );
        iNewEmailPopCount += iCounter;
    }

    iNewEmailImapCount = 0;
    RArray< TImapAccount > imapAccounts( 5 );
    mailAccount->GetImapAccountsL( imapAccounts );
    CleanupClosePushL( imapAccounts );
    for( TInt i = 0; i < imapAccounts.Count(); i++ ) // reading all email inboxes (IMAP)
    {
        TInt iCounter = 0;
        CalcUnreadEntriesL( imapAccounts[i].iImapService, iCounter );
        iNewEmailImapCount += iCounter;
    }
    CleanupStack::PopAndDestroy( 3 ); // imapAccounts popAccounts mailAccount
    iCalcCompleted = ETrue;
} break;

default:
    break;
}
}
```

Internal Links

- [How to get the count of unread messages from Inbox](#)
- [Messaging](#)
- [How to retrieve e-mail contents](#)