CalInterimAPI

The new CalInterim API has been introduced with the Symbian 9.1 releases to replace the old Agenda Model API.

Connecting

The CCalSession is the interface to the Calendar file. The instantiation of CCalSession will result in a connection to the Calendar Server:

Library reqired:

```
LIBRARY calinterimapi.lib
```

Header files:

```
#include <calsession.h>
#include <calprogresscallback.h>
#include <caliterator.h>
#include <calentry.h>
#include <calentryview.h>
#include <caltime.h>
#include <calalarm.h>
#include <calcategory.h>
#include <caluser.h>
#include <calrrule.h>
#include <calinstance.h>
#include <calinstanceview.h>


class CCalendar : public MCalProgressCallBack
    {
    CCalendar();
    void ConstructL();
    ~CCalendar();
    [...]
    private:
    void OpenCalendarL();
    [...]
    CCalSession*       iCalSession;
    CCalIter*       iCalIter;
    CCalEntryView*       iCalEntryView;
    HBufC8*       iNext;
    TBool iReady;
    [...]
    }
```

Source file:

```
#include "Calendar.h"

CCalendar::CCalendar()
    {
    }

CCalendar::~CCalendar()
    {
    if(iCalIter)
        {
 delete iCalIter;
 }
    if(iCalEntryView)
 {
 delete iCalEntryView;
 }
    if(iCalSession)
 {
 delete iCalSession;
 }
    if( iNext )
 {
 delete iNext;
 iNext = NULL;
 }
    }

void CCalendar::ConstructL()
    {
    iReady=EFalse;
    iNext = HBufC8::NewL(50);
    OpenCalendarL();
    }

void CCalendar::OpenCalendarL()
    {
    // allocate and construct server
    // Check that calendar exists, and if not, create it.
    // Calendar does not exist until it is created by calendar app - or by us.
    iCalSession = CCalSession::NewL();

    //Open the default calendar file
    TRAPD(aErr, iCalSession->OpenL(KNullDesC));
    if(aErr == KErrNotFound)
        {
 TRAPD(err, iCalSession->CreateCalFileL(iCalSession->DefaultFileNameL()) );
 TRAPD(aErr2, iCalSession->OpenL(KNullDesC));
 }

    iCalEntryView = CCalEntryView::NewL(*iCalSession, *this);
    iCalIter = CCalIter::NewL(*iCalSession);
```


(C) Copyright Nokia 2013. All rights reserved.

```
    //Reset to the begin of the Calendar entries
    TPtr8 aAddress( iNext->Des() );
    aAddress = iCalIter->FirstL();
    }
```

Do not forget implement other functions!

```
// Called during calendar entry view creation
void CCalendar::Progress(TInt aPercentageCompleted)
    {
    }

void CCalendar::Completed(TInt aError)
    {
    if(aError==KErrNone)
        iReady=ETrue;
    }

// Returns whether or not progress notification is required
TBool CCalendar::NotifyProgress()
    {
    // Progress notification is required
    return ETrue;
    }

// Utility function to handle CCalEntry insert
// Destroy the RPointerArray
void DestroyRPointerArray(TAny* aPtr)
    {
    RPointerArray<CCalEntry>* self = static_cast<RPointerArray<CCalEntry>*> (aPtr);
    self->ResetAndDestroy();
    }
```

Insert new Appointment

To insert new item with a given ID you should create a CCalEntry item and fill out with the desidered data.

```
void CCalendar::NewApptL(TInt aID)
    {
    if(!iReady) //if this function is called too soon then return. Calendar opening
                //must finish first.
    return;

    //Create entry with the given new GUid
    HBufC8* guidBuf;
    TBuf8<30> tmpGuid;
    tmpGuid.Num( aID );
    guidBuf = tmpGuid.AllocL();

    CCalEntry* appt = CCalEntry::NewL(CCalEntry::EAppt, guidBuf,
    CCalEntry::EMethodNone, 0);
    CleanupStack::PushL(appt);

    //Some data
    appt->SetSummaryL( _L("Summary") );
    appt->SetLocationL( _L("Location") );
    appt->SetDescriptionL( _L("Description") );

    //Start / end date
    TTime start;
    start.UniversalTime();
    TTime end;
    end.UniversalTime();
    TCalTime startCalTime;
    startCalTime.SetTimeUtcL(start);
    TCalTime endCalTime; //NullTTime()

    //Comment out the next line if you do not want to set end time
    endCalTime.SetTimeUtcL(end);

    //Set it
    appt->SetStartAndEndTimeL(startCalTime, endCalTime);

    //Store this new Entry

    RPointerArray<CCalEntry> entryArray;
    CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &entryArray));
    entryArray.AppendL(appt);
    TInt success(0);
    iCalEntryView->StoreL(entryArray, success);
    entryArray.Reset();
    CleanupStack::PopAndDestroy(&entryArray);
    CleanupStack::PopAndDestroy(appt);
    }
```

Insert new Todo

To insert new item with a given ID you should create a CCalEntry item and fill out with the desidered data.

```
void CCalendar::NewTodoL(TInt aID)
    {
    //Create an antry with the given GUid
    HBufC8* guidBuf;
    TBuf8<30> tmpGuid;
    tmpGuid.Num( aID );
    guidBuf = tmpGuid.AllocL();

    CCalEntry* todo = CCalEntry::NewL(CCalEntry::ETodo, guidBuf,
     CCalEntry::EMethodNone, 0);
    CleanupStack::PushL(todo);
```


(C) Copyright Nokia 2013. All rights reserved.

```
    todo->SetSummaryL( _L("Summary") );

    //Priority
    TInt priority = 1;
    todo->SetPriorityL(priority);

    //Completed
    TCalTime calTime;
    TTime time; time.UniversalTime();
    calTime.SetTimeUtcL(time);
    todo->SetCompletedL(ETrue, calTime);

    //End date
    TTime end;
    end.UniversalTime();
    TCalTime calEndDate;
    calEndDate.SetTimeUtcL(end);
    todo->SetStartAndEndTimeL(calStartDate, calEndDate);

    //Store New Entry
    RPointerArray<CCalEntry> entryArray;
    CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &entryArray));
    entryArray.AppendL(aTask);
    TInt success(0);
    iCalEntryView->StoreL(entryArray, success);
    entryArray.Reset();
    CleanupStack::PopAndDestroy(&entryArray);
    CleanupStack::PopAndDestroy(todo);
}
```

Add old AgendaModel Alarm data to the new CCalEntry Appointment item

This code sample is useful if you port your application from the previous agenda model. In the old model you should use the SetAlarm(TTimeIntervalDays aDaysWarning,TTimeIntervalMinutes aTime). In the new API you can set the alarm only by giving the offset time according to the start date of the entry.

```
//Get old alarm data
TTimeIntervalDays aDays = ...;
TTimeIntervalMinutes aMinutes= ...;

//tmp variables
TInt oneDayInMin = 1440;
TInt64 oneDayInMicroSec = 86400000000LL;
TInt64 oneMinInMicroSec = 60000000;
TCalTime startCalTime;

//Set the start time
startCalTime.SetTimeUtcL(start);
TInt offSet =
((startCalTime.TimeLocalL().Int64() % oneDayInMicroSec) / oneMinInMicroSec) -
 aMinutes.Int();
if( offSet < 0 ) offSet = oneDayInMin - Abs(offSet);
TTimeIntervalMinutes tmp( offSet + aDays.Int() * oneDayInMin);
CCalAlarm* alarm = CCalAlarm::NewL();

if(alarm)
    {
    alarm->SetTimeOffset(tmp);
    appt->SetAlarmL(alarm);
    delete alarm;
    }
```

Add old AgendaModel Alarm data to the new CCalEntry Todo item

This lines are similar to the appointment code, but now the alarm offset is calculated from the end date (due date).

```
//Get old alarm data
TTimeIntervalDays aDays = ...;
TTimeIntervalMinutes aMinutes= ...;

//tmp variables
TInt oneDayInMin = 1440;
TInt64 oneDayInMicroSec = 86400000000LL;
TInt64 oneMinInMicroSec = 60000000;
TCalTime endCalTime;

//Set end date
endCalTime.SetTimeUtcL(dueDate);
TInt offSet =
((endCalTime.TimeLocalL().Int64() % oneDayInMicroSec) / oneMinInMicroSec) -
 aMinutes.Int();
TTimeIntervalMinutes tmp( offSet + aDays.Int() * oneDayInMin);
CCalAlarm* alarm = CCalAlarm::NewL();
if( alarm )
    {
    alarm->SetTimeOffset(tmp);
    todo->SetAlarmL(alarm);
    delete alarm;
    }
```

Delete entry

```
void CCalendar::DeleteItemL(TInt aID)
    {
    CCalEntry* aEntry = NULL;
    TBuf8<30> tmpGuid;
    tmpGuid.Num( aID );
    HBufC8* guidBuf = tmpGuid.AllocL();
    RPointerArray<CCalEntry> tmpArray;
    CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &tmpArray));
    iCalEntryView->FetchL( guidBuf->Des(), tmpArray );
    delete guidBuf;

    if( tmpArray.Count() == 0 )
```


(C) Copyright Nokia 2013. All rights reserved.

```
        {
 CleanupStack::PopAndDestroy(&tmpArray);
 return;
 }
    aEntry = tmpArray[0];
    iCalEntryView->DeleteL(*aEntry);
    CleanupStack::PopAndDestroy(&tmpArray);
    }
```

Update Appointment entry

Because the CCalEntryView->UpdateL() can't update all the attributes of an entry i use a tmpEntry to store the copied data to store the new modified entry, there are some entry swapping to avoid the data loss if a break occour. You can also implement this function using the CopyCalEntryL() function, described at the end of this document.

```
void CCalendar::UpdateApptL(TInt aID)
    {
    CCalEntry* aAppt = NULL;
    CCalEntry* tmpEntry = NULL;
    TInt aID = GetInt(*aIDText);
    TBuf8<30> tmpGuid;
    tmpGuid.Num(aID);
    HBufC8* guidBuf = tmpGuid.AllocL();
    HBufC8*  guidTmp;

    TBool aFound = EFalse;

    RPointerArray<CCalEntry> array;
    CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &array));
    TRAPD(aErr, iCalEntryView->FetchL(guidBuf->Des(), array) );
    if (!aErr && array.Count() > 0 )
        {
 aAppt = array[0];
 if( aAppt->EntryTypeL() != CCalEntry::EAppt ) aAppt = NULL;
 aFound = ETrue;
 }

    if (aFound)
 {        //This will create new unique UID
 HBufC8* guidTmp = CalenInterimUtils::GlobalUidL();
/*
        //Check the existence of the newly created tmp guid
 TInt itemCount = 0;
 do
     {
     TInt guidNum = Math::Random();
     TBuf8<30> tmpGuid;
     tmpGuid.Num( Abs(guidNum) );
     guidTmp = tmpGuid.AllocL();
     RPointerArray<CCalEntry> tmparray;
     CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &tmparray));
     iCalEntryView->FetchL(guidTmp->Des(), tmparray);
            itemCount = tmparray.Count();
     CleanupStack::PopAndDestroy(&tmparray);
     if( itemCount > 0 ) delete guidTmp;
     }
 while( itemCount > 0 );
*/
 tmpEntry =
  CCalEntry::NewL(CCalEntry::EAppt, guidTmp, CCalEntry::EMethodNone, 0);
 CleanupStack::PushL(tmpEntry);
 CopyCalEntryL(aAppt, tmpEntry);
 CleanupStack::Pop(tmpEntry);
 }

    CleanupStack::PopAndDestroy(&array);
    if( tmpEntry ) CleanupStack::PushL(tmpEntry);

    //Set new data:
    tmpEntry->SetSummaryL(_L("Summary"));
    // [...]

    //Store it

    if( tmpEntry ) CleanupStack::Pop(tmpEntry);
    if (aFound)
        {
 HBufC8* guidUpdate = guidBuf->AllocL();
 CCalEntry* updateEntry =
  CCalEntry::NewL(CCalEntry::EAppt, guidUpdate, CCalEntry::EMethodNone, 0);
 CleanupStack::PushL(updateEntry);
 CopyCalEntryL(tmpEntry, updateEntry);
 CleanupStack::Pop(updateEntry);

 HBufC8* deleteGuidTmp = guidTmp->AllocL();

 //Insert the tmp entry
 RPointerArray<CCalEntry> insertArray;
 CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &insertArray));
 insertArray.AppendL(tmpEntry);
 TInt success(0);
 iCalEntryView->StoreL(insertArray, success);
 insertArray.Reset();
 CleanupStack::PopAndDestroy(&insertArray);
 delete tmpEntry;

 //Delete old
 RPointerArray<CCalEntry> deleteArray;
 CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &deleteArray));
 iCalEntryView->FetchL(guidBuf->Des(), deleteArray);
 CCalEntry* deleteEntry = deleteArray[0];
 iCalEntryView->DeleteL(*deleteEntry);
 CleanupStack::PopAndDestroy(&deleteArray);

 //insert new update Entry
 RPointerArray<CCalEntry> updateArray;
 CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &updateArray));
 updateArray.AppendL(updateEntry);
 success = 0;
 iCalEntryView->StoreL(updateArray, success);
 updateArray.Reset();
 CleanupStack::PopAndDestroy(&updateArray);
 delete updateEntry;
```


(C) Copyright Nokia 2013. All rights reserved.

```
//Delete tmp Entry
RPointerArray<CCalEntry> deleteTmpArray;
CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &deleteTmpArray));
iCalEntryView->FetchL(deleteGuidTmp->Des(), deleteTmpArray);
CCalEntry* deleteTmpEntry = deleteTmpArray[0];
iCalEntryView->DeleteL(*deleteTmpEntry);
CleanupStack::PopAndDestroy(&deleteTmpArray);
delete deleteGuidTmp;
}
   delete guidBuf;
   }
```

Update Todo item

The Todo entry update can be made in similar mode than the Appointment entry.

Convert item CCalEntry IDs to old AgendaModel entry IDs

If you are porting your application from the old Agenda API to the new CalInterim API, presumably you have head-ache how to convert the new alphanumeric IDs to the old integer ID-s. Here is an example:

```
//////////////////////////////////////////////////////////////////////
/*
Because the Global Uid(22 char length &HBufC8) of an entry
is out of range of TInt32...
This function must be called after the & CCalIter creation
and before any other functions.
*/

// Update the Global uids to be in the range of TInt32
void CCalendar::ConvertL()
{
    HBufC8* next = HBufC8::NewL(50);
    TPtr8 aAddress( next->Des() );
    aAddress = iCalIter->FirstL();

    while( next->Des() != KNullDesC8 )
        {
        //Checking if the ( UID is string) || (bigger then 2147483646)
 TBool toConvert = EFalse;
 TBuf8<100> des8 = _L8("");
 TRAPD(err_fetch, des8.Copy( next->Des() ) );
 if( err_fetch == KErrNone )
     {
     TLex8 lex8 = TLex8(des8);
     TInt id = 0;
     TInt err_lex = lex8.Val(id);
     if( err_lex != KErrNone ) toConvert = ETrue;
     }

        //Update
 if( toConvert != EFalse )
             {
     RPointerArray<CCalEntry> fetchArray;
     CleanupStack::PushL(TCleanupItem(DestroyRPointerArray, &fetchArray));
     iCalEntryView->FetchL(next->Des(), fetchArray);
     CCalEntry* aEntry = fetchArray[0];

     ////////////////////////////////////////////////


     //Convert only the Appontment & Todo
     if( aEntry->EntryTypeL() == CCalEntry::EAppt || aEntry->EntryTypeL() ==
                 CCalEntry::ETodo )
                 {
  //Check the existence of an entry with the new GUid
  HBufC8* guidBuf;
         TInt itemCount = 0;
  do
      {
      TInt guidNum = Math::Random();
             TBuf8<30> tmpGuid;
      tmpGuid.Num( Abs(guidNum) );
             guidBuf = tmpGuid.AllocL();
      RPointerArray<CCalEntry> tmparray;
      CleanupStack::PushL(TCleanupItem(DestroyRPointerArray,
                    &tmparray));
      iCalEntryView->FetchL(guidBuf->Des(), tmparray);
      itemCount = tmparray.Count();
      CleanupStack::PopAndDestroy(&tmparray);
      if( itemCount > 0 ) delete guidBuf;
      }
  while( itemCount > 0 );
                    CCalEntry* updateEntry;
      if( aEntry->EntryTypeL() == CCalEntry::EAppt )
          {
          updateEntry =
          CCalEntry::NewL(CCalEntry::EAppt, guidBuf,
                        CCalEntry::EMethodNone, 0);
          }
      else
          {
          updateEntry =
          CCalEntry::NewL(CCalEntry::ETodo, guidBuf,
                        CCalEntry::EMethodNone, 0);
          }

      //Copy: because updateEntry->CopyFromL( *aEntry, EDontCopyId );
                    //doesn't work :(
      CopyCalEntryL(aEntry, updateEntry);

      RPointerArray<CCalEntry> modifyingArray;
      CleanupStack::PushL(TCleanupItem(DestroyRPointerArray,
                    &modifyingArray));
      modifyingArray.AppendL(updateEntry);
      TInt numberOfEntries(0);
      iCalEntryView->StoreL(modifyingArray, numberOfEntries);
             CleanupStack::PopAndDestroy(&modifyingArray);
```


(C) Copyright Nokia 2013. All rights reserved.

```
      //Delete old
      RPointerArray<CCalEntry> deleteArray;
             CleanupStack::PushL(TCleanupItem(DestroyRPointerArray,
                        &deleteArray));
      iCalEntryView->FetchL(aEntry->UidL(), deleteArray);
      CCalEntry* deleteEntry = deleteArray[0];
             iCalEntryView->DeleteL(*deleteEntry);
      CleanupStack::PopAndDestroy(&deleteArray);
      }
         CleanupStack::PopAndDestroy(&fetchArray);
         }
     aAddress = iCalIter->NextL();
     }
        delete next;
 next = NULL;
    }
```

Copy entry to an other one without Global ID

This function is usefull if CCalEntry CopyFromL(*aEntry, EDontCopyId) doesn't work. aSource and aTarget should exist before calling this function. All 2 entries should be the same type!

```
void CCalendar::CopyCalEntryL(CCalEntry* aSource, CCalEntry* aTarget)
    {
    aTarget->SetSummaryL( aSource->SummaryL() );
    aTarget->SetDescriptionL( aSource->DescriptionL() );
    aTarget->SetLastModifiedDateL();
    aTarget->SetDTStampL( aSource->DTStampL() );
    aTarget->SetLocationL( aSource->LocationL() );
    aTarget->SetStatusL( aSource->StatusL() );
    aTarget->SetReplicationStatusL( aSource->ReplicationStatusL() );
    aTarget->SetPriorityL( aSource->PriorityL() );

    //On Nokia E60 - sometimes at Todo entries, getting startTime/endTime Leaves!
    //In this case create a new NullTTime Enty
    TCalTime startCalTime;
    TRAPD(err, startCalTime = aSource->StartTimeL());
    if( err != KErrNone )
        {
 TTime time = Time::NullTTime();
 startCalTime.SetTimeUtcL(time);
 }

    TCalTime endCalTime;
    TRAPD(err1, endCalTime = aSource->EndTimeL());
    if( err1 != KErrNone )
        {
 TTime time = Time::NullTTime();
 endCalTime.SetTimeUtcL(time);
 }

    aTarget->SetStartAndEndTimeL(startCalTime, aSource->EndTimeL());

    if( aSource->EntryTypeL() == CCalEntry::ETodo )
        {
 TCalTime time = aSource->CompletedTimeL();
 if( time.TimeUtcL() != Time::NullTTime() ) aTarget->SetCompletedL
                (ETrue time);
 aSource->CompletedTimeL().TimeUtcL(), ETrue, 3);
 }

    TCalRRule rule;
    if( aSource->GetRRuleL( rule ) != EFalse ) aTarget->SetRRuleL( rule );

    RArray<TCalTime> array;
    aSource->GetRDatesL( array );
    if( array.Count() > 0 ) aTarget->SetRDatesL( array );

    array.Reset();
    aSource->GetExceptionDatesL( array );
    if( array.Count() > 0 ) aTarget->SetExceptionDatesL( array );
    array.Close();

    CCalAlarm* alarm = aSource->AlarmL();
    aTarget->SetAlarmL( alarm );
    if( alarm )
        {
 aSource->AlarmL()->TimeOffset().Int(), 3);
 delete alarm;
 }

    //If,
    //Category, Attendee, Organizer, PhoneOwner, Method,
    //SequenceNumber, RecurrenceID, TzRules, LocalUid are not needed
    }
```

--egeri 09:51, 8 June 2007 (UTC)

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.