

# Archived:Voice call recorder in Symbian C++



**Archived:** This article is [archived](#) because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template `{{ReviewForRemovalFromArchive|user=~~~~|write your reason here}}`.

## Introduction

This article describes how we can record a voice call automatically by 3rd party application. When ever we receive a phone call then there are state change in CTelephony class. If we monitor the state and when the state is changed is detected to CTelephony::EStatusConnected then we can start recording (for example). When the state is changed to CTelephony::EStatusDisconnecting then we can stop recording. Following code example shows how to do this.

```
// We get a notification from telephony subsystem.
void CHelloWorldBasicAppUi::CallStatusChangedL(CTelephony::TCallStatus& aStatus, TInt aError)
{
    if(aError != KErrNone) // Some thing wrong, we should handle though this example doesn't
    {
        return;
    }
    switch(aStatus)
    {
        case CTelephony::EStatusConnected:
        {
            //Recording start by HandleCommandL() method
            HandleCommandL(EProgCmdRecord);
        }
        break;
        case CTelephony::EStatusDisconnecting:
        case CTelephony::EStatusHold:
        {
            //Recording stop by HandleCommandL() method
            HandleCommandL(EProgCmdStop);
        }
        break;
        default:
        break;
    }
}
```

Call status is updated to the previous class by the following code.

```
void CCallMonitor::RunL()
{
    iCallback.CallStatusChangedL(iCurrentStatus.iStatus, iStatus.Int());
    StartListening();
}

void CCallMonitor::StartListening()
{
    Cancel();
    iCurrentStatus.iStatus = CTelephony::EStatusUnknown;
    iTelephony->NotifyChange(iStatus, CTelephony::EVoiceLineStatusChange, iCurrentStatusPckg);
    SetActive();
}
```

We need to create CMdaAudioRecorderUtility class with a priority and preferences as shown by the following code.

```
#define KAudioPriority 80
#define KAudioPreference 0x00060001

iRecorderUtility = CMdaAudioRecorderUtility::NewL(
    *this,
    NULL,
    KAudioPriority,
    (TMdaPriorityPreference)KAudioPreference);
```

## Example Applications

When we receive a call the call is automatically recorded to a file. That file can be played with a player. The application is tested with N95 8GB, complete source code can be found here: [File:CallAudioRecord 31.zip](#)

