

<b>ID</b>	TSS000685	<b>Creation date</b>	June 13, 2007
<b>Platform</b>	S60 3rd Edition	<b>Devices</b>	
<b>Category</b>	Symbian C++	<b>Subcategory</b>	Networking, Telephony

**Keywords (APIs, classes, methods, functions):**

## Overview

Retrieving incoming and outgoing call numbers

## Description

The following code example demonstrates how to retrieve telephone numbers for incoming and outgoing calls using the 3rd party telephony API (CTelephony).

The below code is meant to be implemented in an active object, that is, a class derived from CActive.

## Solution

```
// Class members
CTelephony*           iTelephony;
CTelephony::TCallStatusV1  iLineStatus;
CTelephony::TCallStatus  iLastInformedLineStatus;
Construction & initialization:
iTelephony = CTelephony::NewL();
// Initialize with ?unknown? statuses
iLineStatus.iStatus = CTelephony::EStatusUnknown;
iLastInformedLineStatus = CTelephony::EStatusUnknown;
CTelephony::TCallStatusV1Pckg lineStatusPckg( iLineStatus );
// Request notification about the changes in voice line status
iTelephony->NotifyChange( iStatus, CTelephony::EVoiceLineStatusChange, lineStatusPckg );
SetActive();
Implementation of CActive::RunL()
// Within CActive::RunL():
if( iLineStatus.iStatus == CTelephony::EStatusDialling )
{
    GetNumber();
}
if(iLineStatus.iStatus == CTelephony::EStatusRinging)
{
    if(iLastInformedLineStatus != CTelephony::EStatusDialling)
    {
```

## TSS000685\_-\_Retrieving\_incoming\_and\_outgoing\_call\_numbers

```
        GetNumber();
    }
}
iLastInformedLineStatus = iLineStatus.iStatus;
Implementation of CActive::DoCancel():
iTelephony->CancelAsync( CTelephony::EVoiceLineStatusChangeCancel );
Implementation of GetNumber():
CTelephony::TCallInfoV1 callInfoV1;
CTelephony::TCallInfoV1Pckg callInfoV1Pckg( callInfoV1 );
CTelephony::TCallSelectionV1 callSelectionV1;
CTelephony::TCallSelectionV1Pckg callSelectionV1Pckg( callSelectionV1 );
CTelephony::TRemotePartyInfoV1 remotePartyInfoV1;
CTelephony::TRemotePartyInfoV1Pckg remotePartyInfoV1Pckg( remotePartyInfoV1 );
callSelectionV1.iLine = CTelephony::EVoiceLine;
callSelectionV1.iSelect = CTelephony::EInProgressCall;
iTelephony->GetCallInfo( callSelectionV1Pckg, callInfoV1Pckg, remotePartyInfoV1Pckg );
if( remotePartyInfoV1.iRemoteIdStatus == CTelephony::ERemoteIdentityAvailable )
{
    if( remotePartyInfoV1.iRemoteNumber.iTelNumber.Length() > 0 )
    {
        // Incoming call number can be read from
        // remotePartyInfoV1.iRemoteNumber.iTelNumber;
    }
}
if( callInfoV1.iDialledParty.iTelNumber.Length() > 0 )
{
    // Outgoing call number can be read from
    // callInfoV1.iDialledParty.iTelNumber;
}
```