

## How\_to\_establish\_data\_call\_in\_S60\_3rd\_edition\_devices

Previously in 2nd edition devices, **ETEL** APIs like *RPhone*, *RLine*, *RCall* are used for establishing a CSD call. But ETel APIs are deprecated from 3rd edition onwards and it was replaced by **Etel 3rd Party API** (*CTelephony*).

But even this new ETel3rdparty API has the data call functionality removed as it is Symbian's opinion that data calls do not have a valid use case and that **developers are advised to switch towards using packed data** ( like raw sockets, RTP, ...) for their application's data exchange.

However the ETEL APIs are still present in the 3rd edition public SDKs and can be used for establishing data call. However there are no guarantees that full support is implemented in the platform (or) the entire support for CSD call functionality will be available in future platform releases.

Here are the steps for initiating and answering data call :

1) The preliminary steps are similar to initiate a voice call, like establishing session to the *RTelServer*, opening the *RPhone* instance with the connected server session,etc.. But for getting the line info, it should be checked for the existence of DataCall capability. Here is the code snippet:

```
RCall iCall;
RLine iLine;
RTelServer::TPhoneInfo phoneInfo;
RTelServer iServer;
TInt iNumberLines;
RPhone iPhone;
RPhone::TLineInfo lineInfo;
iServer.Connect();
if (iServer.LoadPhoneModule(_L("phonetsy")) != KErrNone)
{
return;
}
iServer.GetPhoneInfo(0, phoneInfo);
iPhone.Open(iServer, phoneInfo.iName);
iPhone.EnumerateLines(iNumberLines);
for (TInt a = 0; !FoundLine && a < iNumberLines; a++)
{
iPhone.GetLineInfo(a, lineInfo);
if (lineInfo.iLineCapsFlags & RLine::KCapsData)
FoundLine = ETrue;
}
iLine.Open(iPhone, lineInfo.iName);
```

2) If you want to listen for incoming data call, then the following code snippets demonstrates the steps for doing it( which is somewhat similar to voice call steps):

```
TName call_name;
// asynchronous request waiting for incoming call notification
iLine.NotifyIncomingCall(iStatus, call_name);
SetActive();
```

## How\_to\_establish\_data\_call\_in\_S60\_3rd\_edition\_devices

After the asynchronous request completes, the call should be opened with the existing line as follows:

```
iCall.OpenExistingCall(iLine, call_name);
```

Then the opened call should be answered as follows:

```
iCall.AnswerIncomingCall(iStatus);  
SetActive();
```

Once this answering call request is completed, the commport should be used for loaning the data port as follows:

```
RCall::TCommPort iCommPort;  
iCall.LoanDataPort(iStatus, iCommPort);  
SetActive();
```

The completion of this request indicates that the dataport loaning is over. Now the data call is completely established and ready for data transfer. For transferring data through the CSD call, a session is established with the *Commserver* and the appropriate comm module should be loaded. Then the data can be received/transferred using *RComm* APIs as follows:

```
RCommServ iCommServer;  
RComm iComm;  
iCommServer.Connect();  
iCommServer.LoadCommModule(iCommPort.iCsy);  
TCommConfig config;  
iComm.Open(iCommServer, iCommPort.iPort, ECommShared);  
iComm.Config(config);  
config().iTerminatorCount = 0;  
iComm.SetConfig(config); // for setting configurations  
  
iComm.Read() // should be used for receiving data from the CSD call  
iComm.Write() // should be used for sending data through the CSD call
```

3) For initiating data call, a new call should be opened using *RCall* and the destination number is dialed using *Dial()* API. Once the dial is successful, then the data port should be loaned as usual for further data transfer through the established call.

## How\_to\_establish\_data\_call\_in\_S60\_3rd\_edition\_devices

```
TInt ret;
iCall.OpenNewCall(iLine, call_name);
ret = iCall.Dial(_L("+358408033493"));
if (ret != KErrNone)
{
// dialing successful
}
```

This data call functionality seems to have a dependency on *Network mode* as it works properly on *GSM mode* alone.

[KIS000999 - Error -8072 while trying to answer a data call on S60 3rd Edition FP1 devices](#)