

Send data using RTP/RTCP API on Symbian

 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}}`.

The article is believed to be still valid for the original topic scope.

Overview

The RTP/RTCP API available in the SDK API Plug-in pack can be used for delivering different types of data between devices. The Real-time Transport Protocol(RTP) is used to send or receive the data and the RTP Control Protocol(RTCP) provides out-of-band statistics and control information for an RTP flow.

Description

This article demonstrates how to enable RTCP and register an observer for receiving the Receiver report(RR) packet in an RTP session. The receiver report informs the sender about the quality of service. The other RTCP messages that can be exchanged are Sender report(SR), Source description(SDES), End of participation (BYE), and Application-specific message (APP).

Since RTCP streaming is used during an RTP flow, more information on starting a RTP session is given in the following link: [S60 RTP/RTC API](#)

Solution

Header Files:

```
#include <rtpapi.h>
#include <rtpheader.h>
#include <RtpDef.h>
```

Link Against:

```
LIBRARY rtpservice.lib
```

Enabling RTCP and registering an observer in an RTP session

```
iRtpId = iRtpSession->CreateSessionL( sessionParam,
                                     localPort,
                                     ETrue, //enable RTCP
                                     NULL ); //Set default RTCP parameters

// Set remote RTCP address
TInt err1 = iRtpSession->SetRemoteRtcpAddress(iRtpId, addr2 );<br>
//Register for RTCP callback
err1 = iRtpSession->RegisterRtcpObserver(iRtpId, *this);
```

Sending a Receiver Report(RR) to sender:

```
iRtpSession->SendRtcpRrPacket(aStreamId); //aStreamId is the TRtpId
                                         //of a received RTP packet.
```

Receiving the RTCP packet:

```
void CRTPEngine::RrReceived( TRtpId aStreamId, TRtpSSRC aSSRC )  
{  
    //Log the received parameters  
}
```

Unregistering the RTCP callback and closing the session:

```
iRtpSession->UnregisterRtcpObserver(iRtpId);  
iRtpSession->CloseSession(iRtpId);
```

Example Application

This application sends data packets using RTP and receives feedback about the sent packets using RTCP. [File:RTPTTest.zip](#)