
Posted by [Cleo](#) on Sat, 21 Oct 2023 09:27:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

TMemoryStream/TFileStream.

var

WaveIn: HWaveIn;
WaveFormat: TWaveFormatEx;
WaveHeader: TWaveHdr;
Buffer: array[0..1023] of Byte;
FileStream: TFileStream;

procedure TForm1.Button1Click(Sender: TObject);

begin

WaveFormat.wFormatTag := WAVE_FORMAT_PCM;
WaveFormat.nChannels := 1;
WaveFormat.nSamplesPerSec := 44100;
WaveFormat.nAvgBytesPerSec := WaveFormat.nSamplesPerSec * WaveFormat.nChannels *
2;
WaveFormat.nBlockAlign := 2;
WaveFormat.wBitsPerSample := 16;
WaveFormat.cbSize := 0;

WaveInOpen(@WaveIn, WAVE_MAPPER, @WaveFormat, 0, 0, 0);

WaveInStart(WaveIn);

```

WaveInPrepareHeader(WaveIn, @WaveHeader, SizeOf(WaveHeader));

WaveInAddBuffer(WaveIn, @WaveHeader, SizeOf(WaveHeader));
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  FileStream := TFileStream.Create('c:\wav\sound.wav', fmCreate);
  try
    WaveInUnprepareHeader(WaveIn, @WaveHeader, SizeOf(WaveHeader));
    WaveInStop(WaveIn);
    WaveInClose(WaveIn);

    FileStream.WriteBuffer(WaveFormat, SizeOf(WaveFormat));
    FileStream.WriteBuffer(WaveHeader.lpData^, WaveHeader.dwBufferLength);

  finally
    FileStream.Free;
  end;
end;

```

```

var
  WaveIn: HWAVEIN;
  WaveHeader: PWAVEHDR;
  Stream: TMemoryStream;

```

```

//start button
var

```

```

WaveFormat: TWaveFormatEx;

begin
  WaveIn := 0;
  WaveHeader := nil;
  Stream := TMemoryStream.Create;

  WaveFormat.wFormatTag := WAVE_FORMAT_PCM;
  WaveFormat.nChannels := 1;
  WaveFormat.nSamplesPerSec := 44100;
  WaveFormat.wBitsPerSample := 16;
  WaveFormat.nBlockAlign := WaveFormat.nChannels * (WaveFormat.wBitsPerSample div 8);
  WaveFormat.nAvgBytesPerSec := WaveFormat.nSamplesPerSec * WaveFormat.nBlockAlign;
  WaveFormat.cbSize := 0;
  waveInOpen(@WaveIn, WAVE_MAPPER, @WaveFormat, 0, 0, WAVE_FORMAT_DIRECT);

  GetMem(WaveHeader, SizeOf(WAVEHDR));
  WaveHeader^.lpData := AllocMem(8192);
  WaveHeader^.dwBufferLength := 8192;
  WaveHeader^.dwBytesRecorded := 0;
  WaveHeader^.dwUser := 0;
  WaveHeader^.dwFlags := 0;
  WaveHeader^.dwLoops := 0;
  WaveHeader^.lpNext := nil;
  WaveHeader^.reserved := 0;

  waveInPrepareHeader(WaveIn, WaveHeader, SizeOf(WAVEHDR));
  waveInAddBuffer(WaveIn, WaveHeader, SizeOf(WAVEHDR));
  waveInStart(WaveIn);

  //stop button

  waveInStop(WaveIn);
  waveInUnprepareHeader(WaveIn, WaveHeader, SizeOf(WAVEHDR));
  Stream.Write(WaveHeader^.lpData^, WaveHeader^.dwBytesRecorded);
  Stream.SaveToFile('sound.wav');
  FreeMem(WaveHeader^.lpData);
  FreeMem(WaveHeader);
  waveInClose(WaveIn);
  Stream.Free;

```

uses

```

HWaveIn, WaveIO;

var
  WaveIn: THWaveIn;
  WaveFile: TWaveFile;
  WaveFormat: TWaveFormatEx;
  Buffer: Pointer;
  BufferSize: Cardinal;
begin

  WaveIn := THWaveIn.Create(nil);

  WaveIn.WaveFormat.FormatTag := WAVE_FORMAT_PCM;
  WaveIn.WaveFormat.Channels := 1;
  WaveIn.WaveFormat.SamplesPerSec := 48000;
  WaveIn.WaveFormat.BitsPerSample := 16;
  WaveIn.WaveFormat.BlockAlign := WaveIn.WaveFormat.Channels *
(WaveIn.WaveFormat.BitsPerSample div 8);
  WaveIn.WaveFormat.BytesPerSec := WaveIn.WaveFormat.SamplesPerSec *
WaveIn.WaveFormat.BlockAlign;
  WaveIn.BufferSize := 1024;
  WaveIn.Open;

  WaveFormat := WaveIn.WaveFormat;
  WaveFile := TWaveFile.Create('C:\MyRecord.wav', WaveFormat);

  WaveIn.Start;

  while WaveIn.GetBuffer(Buffer, BufferSize) = 0 do
  begin
    WaveFile.WriteData(Buffer, BufferSize);
  end;

  WaveIn.Stop;

  WaveFile.Free;
  WaveIn.Free;
end.

```