A general purpose Base64 decoding routine using Indy

I cobbled together this little routine to use the Indy Component (v10) to decode some base-64 encoded information. I chose the *TBytes* return type because base-64 encoding can be used for any sequence of "octets". I thought it better to not return a string since that implies text was encoded. We can easily return a string from the byte array.

As well as *TBytes* I'm using the relatively new *TBytesStream* stream class, so you will need to make some changes if using older Delphis.

```
function Base64Decode(const EncodedText: string): TBytes;
var
  DecodedStm: TBytesStream;
  Decoder: TIdDecoderMIME;
begin
  Decoder := TIdDecoderMIME.Create(nil);
  try
    DecodedStm := TBytesStream.Create;
    try
      Decoder.DecodeBegin(DecodedStm);
      Decoder.Decode(EncodedText);
      Decoder.DecodeEnd;
      Result := DecodedStm.Bytes;
    finally
      DecodedStm.Free;
    end;
  finally
    Decoder.Free;
  end;
end;
```

Author:	*Peter Johnson*
Contributor:	Jeff DeVore
Added:	2010-03-12
Last updated:	2010-03-12

Copyright © Peter Johnson (*DelphiDabbler*) 2002-2018