Base32

Base32 beschreibt ein Verfahren zur Kodierung von Binärdaten in eine Zeichenfolge, die nur aus 32 verschiedenen ASCII-Zeichen (A..Z, 2..7) besteht, plus einem zusätzlichen Zeichen (=) zum Füllzeichen am Datenende.

Da ich bisher keine ordentlich funktionierende Pascal Variante im Netz finden konnte, nun meine Varianten mit unterschiedlichen Übergabemöglichkeiten.

base32.pas (/code/base32.pas?mode=download) Pascal (3,63 kByte) 21.09.2015 00:28

```
//
*******************************************************************************

//  Title.............. :  base32
//
//  Modulname ......... :  base32.pas
//  Type .............. :  Unit
//  Author ............ :  Udo Schmal
//  Development Status  :  20.09.2015
//  Operating System .. :  Win32/Win64
//  IDE ............... :  Delphi & Lazarus
//
*******************************************************************************

unit base32;
{$ifdef fpc}
  {$mode objfpc}
{$endif}
{$H+}
interface

uses
  SysUtils, Classes;

function Base32Encode(source: string): string;
function Base32Encode(const buf; len: integer): string;
function Base32Encode(var Stream: TMemoryStream): string;
function Base32Decode(source: ansistring): ansistring;

implementation

const
  base32chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ234567';

function Base32Encode(source: string): string;
var
  i: integer;
  nr: int64;
begin
  result := '';
  while length(source) > 0 do
```

```
  begin
    nr := 0;
    for i := 1 to 5 do
    begin
      nr := (nr shl 8);
      if length(source)>=i then
        nr := nr + byte(source[i]);
    end;
    for i := 7 downto 0 do
      if ((length(source)<2) and (i<6)) or
         ((length(source)<3) and (i<4)) or
         ((length(source)<4) and (i<3)) or
         ((length(source)<5) and (i<1)) then
      result := result + '='
    else
      result := result + base32chars[((nr shr (i*5)) and $1F)+1];
    delete(source, 1, 5);
  end;
end;

function Base32Encode(const buf; len: integer): string;
var
  i: integer;
  nr: int64;
  arr: array of byte;
begin
  result := '';
  SetLength(arr, len);
  Move(buf, arr[0], len);
  while length(arr) > 0 do
  begin
    nr := 0;
    for i := 1 to 5 do
    begin
      nr := (nr shl 8);
      if length(arr)>=i then
        nr := nr + byte(arr[i-1]);
    end;
    for i := 7 downto 0 do
      if ((length(arr)<2) and (i<6)) or
         ((length(arr)<3) and (i<4)) or
         ((length(arr)<4) and (i<3)) or
         ((length(arr)<5) and (i<1)) then
      result := result + '='
    else
      result := result + base32chars[((nr shr (i*5)) and $1F)+1];
    if length(arr)>5 then
    begin
      move(arr[5], arr[0], length(arr)-5);
      setlength(arr, length(arr)-5);
```

```
    end
    else
      setlength(arr, 0);
  end;
end;

function Base32Encode(var Stream:TMemoryStream): string;
var
  i: integer;
  nr: int64;
  arr: array of byte;
begin
  result := '';
  Stream.Position := 0;
  SetLength(arr, Stream.Size);
  Stream.Read(arr[0], Stream.Size);
  while length(arr) > 0 do
  begin
    nr := 0;
    for i := 1 to 5 do
    begin
      nr := (nr shl 8);
      if length(arr)>=i then
        nr := nr + byte(arr[i-1]);
    end;
    for i := 7 downto 0 do
      if ((length(arr)<2) and (i<6)) or
         ((length(arr)<3) and (i<4)) or
         ((length(arr)<4) and (i<3)) or
         ((length(arr)<5) and (i<1)) then
      result := result + '='
    else
      result := result + base32chars[((nr shr (i*5)) and $1F)+1];
    if length(arr)>5 then
    begin
      move(arr[5], arr[0], length(arr)-5);
      setlength(arr, length(arr)-5);
    end
    else
      setlength(arr, 0);
  end;
end;

function Base32Decode(source: ansistring): ansistring;
var
  i, p: integer;
  nr: Int64;
begin
  result := '';
  while length(source) >= 8 do
```

```
  begin
    nr := 0;
    for i := 1 to 8 do
    begin
      nr := (nr shl 5);
      p := pos(source[i], base32chars);
      if p>0 then
        nr := nr + p-1;
    end;
    for i := 4 downto 0 do
      result := result + chr((nr shr (i*8)) and $FF);
    delete(source, 1, 8);
  end;
end;

end.
```

Autor: Udo Schmal (#udo.schmal) , veröffentlicht: 19.09.2015, letzte Änderung: 06.09.2023