



Tech Info Library

Pascal: Dollar amount formatter (2 of 2)

```
procedure DOLLAR_FORMAT (var Sample: string);
  var DollarStr, CentStr      : string;
      Where, Dollars, Cents   : integer;

procedure ROUND_CENTS (var Dollars, Cents: string);
  var HowLong, Count: integer;
      Sample        : string;
begin
  Sample:= concat ('0', Dollars, Cents); {Leading zero}
                                           {for carry}

  HowLong:= length (Sample);
  if (Sample [HowLong] > '4') then
    Sample [HowLong - 1]:=
      chr(ord (Sample [HowLong - 1]) + 1);
  for Count:= HowLong downto 1 do begin
    if (Sample [Count] > '9') then begin
      Sample [Count]:= '0';
      Sample [Count - 1]:=
        chr(ord (Sample [Count - 1]) + 1)
    end;
  end;
  while (Sample [1] = '0') do begin
    delete (Sample, 1, 1); {Delete leading zeroes}
    HowLong:= HowLong - 1
  end;
  Sample:= copy (Sample, 1, (HowLong - 1));
  HowLong:= HowLong - 1; {Drop 3rd of 3 place cents}
  Dollars:= copy (Sample, 1, (HowLong - 2));
  Cents:= copy (Sample, (HowLong - 1), 2)
end; {Round_Cents}

begin
  Where:= pos ('.', Sample); {Find decimal}
  if (Where = 0) then begin
    Sample:= concat (Sample, '.');
    Where:= length (Sample)
  end;
  DollarStr:= copy (Sample, 1, (Where - 1));
  CentStr:=
    copy (Sample, (Where + 1), (length (Sample) - Where));
  while (length (CentStr) > 3) do
    delete (CentStr, (length (CentStr)), 1);
```

```

    case (length (CentStr)) of
      0: CentStr:= concat (CentStr, '000');
      1: CentStr:= concat (CentStr, '00');
      2: CentStr:= concat (CentStr, '0')
    end; {Round_Cents needs 3 places}
    ROUND_CENTS (DollarStr, CentStr);
    Sample:= concat (DollarStr, '.', CentStr)
end; {Dollar_Format}

```

```

procedure VAL (RealStr: string; var RealNum: real);
var HowLong, NumDigits,
    Count, Digit, Power: integer;
    Dollars, Cents      : string;
    Number              : real;
begin
  RealNum:= 0; {Beginning default}
  HowLong:= length (RealStr);
  Dollars:= copy (RealStr, 1, (HowLong - 3));
  Cents:= copy (RealStr, (HowLong - 1), 2);
  NumDigits:= length (Dollars);
  Power:= 0; {Beginning default}
  if (NumDigits > 0) then
    for Count:= NumDigits downto 1 do begin
      Digit:= ord (Dollars [Count]) - 48; {Convert Ascii}
                                             {to Decimal}

      Number:= Digit * PwrOfTen (Power);
      RealNum:= RealNum + Number;
      Power:= Power + 1
    end;
    Number:=
      (((ord(Cents [1])-48)*10)+(ord(Cents [2])-48))/100;
    RealNum:= RealNum + Number
  end; {Val}

```

```

begin
  gotoxy (HPos, VPos);
  GET_REAL_STR (Limit, RealStr);
  DOLLAR_FORMAT (RealStr);
  Space:= ''; {Beginning default}
  Limit:= Limit + 2; {Number has cents now}
  for Count:= length (RealStr) to Limit do
    Space:= concat (Space, ' ');
  gotoxy (HPos, VPos);
  write (Space, RealStr); {Writes over input data}
  VAL (RealStr, Value);
  GET_DOLLARS:= Value
end; {Get_Dollars}

```

Apple Technical Communications

Keywords: <None>

=====

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 320