

I experienced the same problem when trying to print source code but also when trying to use either of the standard printer-dialoges components as well as the quickreport-preview fucntion.

As this was kind of frustrating (i have a deadline) i began to debug the delphi code and after some lurking about i found a solution that made it work again.

The problem seemed to occurr in printers.pas (VCL) in the function TPrinter.GetPrinter.

2 consecutive calls are made to EnumPrinters(), the first call to find out how big a buffer is required (i assume) and then the second call is made to fetch the information (after allocating a buffer) - but the second call FAILED (dont know why).

As far as i can guess it's a Windows 95 version problem.....

I made these changes in the function TPrinter.GetPrinters: TStrings; The code changes only affects Windows95, Windows NT will work as before.

NOTE ! I take no responsibility for any problems that might occurr as a result of these changes - the code is provided as is.

halfway down are these lines :

```
begin
  Flags := PRINTER_ENUM_LOCAL;
  Level := 5;
end;
```

change them to

```
begin
  Flags := PRINTER_ENUM_LOCAL;
  Level := 2;
end;
```

a bit further down are these lines

```
with PPrinterInfo5(PrinterInfo)^ do
begin
  LineCur := pPortName;
```

change it to

```
with PPrinterInfo2(PrinterInfo)^ do
begin
  LineCur := pPortName;
```

Save the file to the lib-directory and recompile you library, restart Delphi and try it !

Check out the documentation for EnumPrinter for more information

## The following changes were made but without having any effect

If you are using WinNT then I have seen the same problems and have patched PRINTERS.PAS to correct the problem. It shows itself when you have more than one network printer defined. The problem starts from the fact that WINSPOOL.PAS defines a Printerinfo structure for use with WinNT that does not provide the installed Driver and Port information. When you select a printer using the default PrinterSetupDialog it calls Setprinter which then retrieves the driver and port data. The bug appears in the Tprinter.IsEqual() function which compares the printer name and port to determine if the selected printer is in the current printers list and if it is not it adds it to the list. This is the error in the logic since the printer is in the printers list but the port data is not filled in and causes the function to fail and a second entry is added to the printers array for the updated info.

The error you are seeing is in the code for GetPrinter that searches the printers array and finds the first entry without the Driver and Port filled in and then it fails to properly initialize the printer and you get a invalid devmode structure.

Add the following changes to PRINTERS.PAS:

```
In procedure TPrinter.SetPrinter()
.....
....
  with Printers do // <- this rebuilds the FPrinters list
    for I := 0 to Count - 1 do
      begin
        if TPrinterDevice(Objects[I]).IsEqual(ADriver, ADevice, APort) then
          begin
            J := I;
            Break;
          end
        else
          if ((AnsiComparetext(Strings[I],ADevice) = 0)
            and (TPrinterDevice(Objects[I]).Port = '')) then
            begin
              TPrinterDevice(Objects[I]).Port := APort;
              TPrinterDevice(Objects[I]).Driver := ADriver;
              J := I;
              Break;
            end;
          end;
        .....
        .....
```

Then in SetToDefaultPrinter make the following changes:

```
procedure TPrinter.SetToDefaultPrinter;
var
  I: Integer;
  DefaultPrinter: array[0..79] of Char;
  Cur, ADevice: PChar;
  ADriver, APort: string;
begin
  GetProfileString('windows', 'device', '', DefaultPrinter,
```

```

    SizeOf(DefaultPrinter) - 1);
Cur := DefaultPrinter;
ADevice := FetchStr(Cur);
ADriver := StrPas(Cur);
if pos(',',ADriver)>0 then
begin
    APort := ADriver;
    while (pos(',',APort)>0) do
        system.delete(APort,1,1);
    Setlength(ADriver,pos(',',ADriver)-1);
end
else
begin
    ADriver := '';
    APort := '';
end;
with Printers do
for I := 0 to Count-1 do
begin
    if TPrinterDevice(Objects[I]).Device = ADevice then
begin
    with TPrinterDevice(Objects[I]) do
begin
    if ((APort<>'') and (Port=''))
and ((ADriver<>'') and (Driver='')) then
begin
        Port := APort;
        Driver := ADriver;
    end;
    SetPrinter(PChar(Device), PChar(Driver), PChar(Port), 0);
end;
    Exit;
end;
end;
RaiseError(SNoDefaultPrinter);
end;

```