How to save and load printer settings to / from the registry

> ***QUESTION***
>
> I'm trying to get Windows to remember settings - in particular paper type, size and orientation - configured using the Printer Setup dialogue. It's confusing me. It seems that sometimes Windows remembers these settings and sometimes not. The technique I'm considering is to read and save (to an INI file or wherever) the *Printer.PageHeight*, *Printer.PageWidth* etc. values for the particular printer concerned, so they're available to be loaded for subsequent sessions. What better solutions would people suggest? I can't find a way to get the name of the currently selected paper layout (e.g. Letter, Legal, etc.). It's not among the properties of TPrinter. Anyone have a way to get / set this? I'm working with D5 pro (D6 soon), developing in Win2K, deploying to various flavours from Win'9X to XP.

Answer 1

Windows remembers only changes made to the default settings, using the printer icon in the control panels printers applet. A solution for you would be to save the printers devicemode record, the public part at least:

```
uses
  printers, registry;

{$R *.dfm}

procedure SaveCurrentPrinterSettings(reg: TRegistry);
var
  Device, Driver, Port: array[0..80] of Char;
  DevMode: THandle;
  pDevmode: PDeviceMode;
begin
  Printer.GetPrinter(Device, Driver, Port, DevMode);
  if Devmode <> 0 then
  begin
    {lock it to get pointer to DEVMODE record}
    pDevMode := GlobalLock(Devmode);
    if pDevmode <> Nil then
    try
      reg.WriteBinaryData(device, pDevmode^, sizeof(TDevicemode));
    finally
      {unlock devmode handle.}
      GlobalUnlock(Devmode);
    end;
  end;
end;

procedure LoadCurrentPrinterSettings(reg: TRegistry);
var
  Device, Driver, Port: array[0..80] of Char;
  DevMode: THandle;
  pDevmode: PDeviceMode;
begin
  Printer.GetPrinter(Device, Driver, Port, DevMode);
  if Devmode <> 0 then
  begin
    {lock it to get pointer to DEVMODE record}
    pDevMode := GlobalLock(Devmode);
    If pDevmode <> Nil then
    try
      reg.ReadBinaryData(device, pDevmode^, sizeof(TDevicemode));
    finally
      {unlock devmode handle.}
      GlobalUnlock(Devmode);
    end;
```

```
  end;
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  reg: TRegistry;
begin
  reg:= TRegistry.Create(KEY_READ OR KEY_WRITE);
  try
    reg.RootKey := HKEY_CURRENT_USER;
    if reg.OpenKey('Software\PB\Test', true) then
    try
      SaveCurrentPrinterSettings(reg);
    finally
      reg.CloseKey;
    end
  finally
    reg.free
  end;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  reg: TRegistry;
begin
  reg:= TRegistry.Create(KEY_READ);
  try
    reg.RootKey := HKEY_CURRENT_USER;
    if reg.OpenKey('Software\PB\Test', false) then
    try
      LoadCurrentPrinterSettings(reg);
    finally
      reg.CloseKey;
    end
  finally
    reg.free
  end;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  printdialog1.execute
end;
```

Tip by Peter Below

Answer 2

Check out the Delphi Win API help for *GetPrinter*, *PRINTER_INFO_2*, and *DEVMODE*. I use the following code to read the current info for the printer.

I'm attempting to find some code to set the values, but I'm having problems with user rights in Win2000 Pro. Use the Delphi select printer dialog to set the current printer, otherwise the code will display the settings of the default printer if one is set.

```
function GetPrinterName: String;
var
  prnIndex: Integer;
  prnList: TStrings;
begin
  result := '';
  prnIndex := Printers.Printer.PrinterIndex;
  prnList  := Printers.Printer.Printers;
  if (prnIndex > - 1) and (prnIndex < prnList.Count) then
  begin
    if (prnList.Objects[prnIndex] <> nil) then
      result := TPrinterDevice(prnList.Objects[prnIndex]).Device
    else
      result := prnList[prnIndex];
  end;
end;
```

```
procedure ShowPrinterInfo;
const
  LEVEL_2 = 2;
var
  printerHandle: THandle;
  printerInfo: TPrinterInfo2;
  bytesNeeded: Cardinal;
  printerDefaults: TPrinterDefaults;
  prnName: String;
  pBuffer: Pointer;
  gpResult: Boolean;
  devInfo: TDevMode;
  sMsg: String;
begin
  printerHandle := 0;
  ClearRecord(printerInfo, SizeOf(TPrinterInfo2));
  ClearRecord(printerDefaults, SizeOf(TPrinterDefaults));
  prnName := GetPrinterName;
  if not OpenPrinter(
    PChar(prnName), printerHandle, @printerDefaults
  ) then
    Exit;
  try
    gpResult := WinSpool.GetPrinter(
      printerHandle,
      LEVEL_2,
      @printerInfo,
      SizeOf(printerInfo),
      @bytesNeeded
    );
    if not gpResult and (bytesNeeded > SizeOf(PrinterInfo)) then
    begin
      {get printer call failed because buffer passed was too small}
      GetMem(pBuffer, bytesNeeded);  {allocate buffer of needed size}
      try
        {protect buffer memory allocate}
        {call GetPrinter() again with new, bigger buffer}
        gpResult := WinSpool.GetPrinter(
          PrinterHandle,
          LEVEL_2,
          pBuffer,
          bytesNeeded,
          @bytesNeeded
        );
        if gpResult then
          {success second time}
          {1st bytes are info record, copy values}
          printerInfo := PPrinterInfo2(pBuffer)^;
      finally
        FreeMem(pBuffer);  {deallocate buffer}
      end;
    end;
    if gpResult then
    begin
      sMsg := '';
      devInfo := printerInfo.pDevMode^;
      if ((DM_PAPERSIZE and devInfo.dmFields) > 0) then
        sMsg := 'Paper Size: ' +
          PaperSizeToStr(devInfo.dmPaperSize) + #13;
      if ((DM_PAPERLENGTH and devInfo.dmFields) > 0) then
        sMsg := sMsg + 'Paper Length: ' +
          IntToStr(devInfo.dmPaperLength) + #13;
      if ((DM_PAPERWIDTH and devInfo.dmFields) > 0) then
        sMsg := sMsg + 'Paper Width: ' +
          IntToStr(devInfo.dmPaperWidth) + #13;
      ShowMessage(sMsg);
    end;
  finally
    ClosePrinter(printerHandle);
  end;
end;
```

Tip by David Beardwood

Original resource:	The Delphi Pool
Author:	Peter Below & David Beardwood
Added:	2009-09-14
Last updated:	2009-09-14