

What is BSPrinter?

It is a component whose main function is to provide **print preview** capability to Visual Basic 6.

We could say that it adds print preview to VB6's **Printer object**.

How does it work and how to use it?

1) It replaces the original **Printer object** with a new object that is 100% compatible with it, a direct replacement.

To do so you need to add the following code to your program:

```
Public Property Get Printer() As Printer
    Set Printer = PrinterReplacement
End Property

Public Property Set Printer(nPrinter As Printer)
    Set PrinterReplacement = nPrinter
End Property
```

You can place this code in a **bas module** or in the form. If you place it in a bas module it will serve for all the project, but if you do in the form, you should change the declaration to Private and it will be useful only for that form.

2) It also has a control, the **PrintPreview control**.

In order to make it work, you need to put the printing code inside the **PrepareReport event** of one of these controls, or to call the print procedure from there.

```
Private Sub PrintPreview1_PrepareReport(Cancel As Boolean)
    Printer.Print "Hello World."
    ' more printing code
End Sub
```

3) Call the **ShowPreview** method of the PrintPreview control.

```
PrintPreview1.ShowPreview
```

Instead of calling the print procedure directly, now you call the ShowPreview method of the PrintPreview control (from the command button, toolbar button, menu option or from where you previously called the print procedure).

That's it, you should now have print preview in your program by doing these 3 simple steps.

Considerations

- 1) Your code should be prepared to handle any paper size, by taking into account the page size from the **Printer.ScaleWidth** and **Printer.ScaleHeight** properties, and not assuming certain fixed size like Letter (for instance).
- 2) Now your code doesn't need to care about margins, that's automatically handled by the component.
Printer.ScaleWidth and **Printer.ScaleHeight** are all writable space now. If your code handles margins, you'll have to remove that from your code.
- 3) You don't need to care about **page numbers**, they are now automatically handled by the component too. If your printing routine adds page numbers, you should remove that from the code.
- 4) If you don't want to remove your margins and page numbers, you can make the print preview work, you need to set these properties:

```
PrintPreview1.HandleMargins = False  
PrintPreview1.PrintPageNumbers = False  
PrintPreview1.AllowUserChangePageNumbers = False
```

The user will lose the ability to change margins and page numbers properties from the component's user interface, but your existing code can work without having to do these changes.

That is shown in Sample 6 of the sample projects.

- 5) If your form is shown modally, you'll need to declare the PrintPreview control event procedures as **Public** instead of **Private**.

More Information

It can print the MSFlexgrid/ MSHFlexgrid with a single line of code, with the method **PrintGrid**.

You could also use it to print any **RecordSet**, assigning it to the DataSource of a grid that does not need to be visible.

Together with this document should be another one that has all the property, method, and event **references** explained. Also constants and other component objects.

The online version is at www.balkesoft.com/bsprinter/docs/reference

The **user interface** (of the dialog boxes, such as the Print Preview dialog box) has built-in support for several **languages**. In the current version it supports English, Spanish, French, German, Italian, Portuguese, Greek, Chinese, Hebrew and Arabic.

Support for other languages can be added as shown on Sample 26 (under 'Advanced samples 2').

For languages that are not supported, English will be used.

The usage is very simple and straightforward (most of the times), just add a PrintPreview control to the form and do the steps listed above. But it still allows many customizations and to do some special things. Most of them are covered in the **sample projects**.

You don't need to learn all this to start using it, but if you want to know how to do something, the samples are there to see how.

The sample projects list is:

- 1 - Hello World
- 2 - Positioning on page
- 3 - Print FlexGrid
- 4 - Print RichTextBox
- 5 - Save PDF file
- 6 - Full backward compatibility
- 7 - Show more pages by default
- 8 - Print using PrintEx method (for printing Unicode and more)
- 9 - Allow or disallow user actions
- 10 - Show margins
- 11 - Show Print and Page setup from the form menu
- 12 - Remember user preferences
- 13 - Set a Footer or Header on every page
- 14 - Print watermarks
- 15 - Print form and controls
- 16 - Use the Auxiliary button
- 17 - Set the starting page number to other than 1
- 18 - Page numbers settings
- 19 - Change default margins
- 20 - Change the default grid style when printing a grid
- 21 - Customize User Interface look
- 22 - Save PDF from a toolbar button
- 23 - Advanced Footer and Header printing
- 24 - Report with dynamic table of contents
- 25 - API Drawing
- 26 - Translate the user interface to another language
- 27 - Use in Class modules
- 28 - Global event handler

SxS Installations

This is the text to add to the manifest file in SxS installations:

The component itself is ready to handle themed controls, DPI aware, DPI aware per monitor, DPI aware per monitor v2. But these features depend on how the host program is manifested.

It is also ready to handle non-integer twips-per-pixel settings, different system font sizes, high contrast themes, Unicode languages and right-to-left languages on its user interface.

The component has no dependencies, only the ocx file needs to be distributed with your application.