



Module 6: Printing and Reporting in Windows Forms Applications

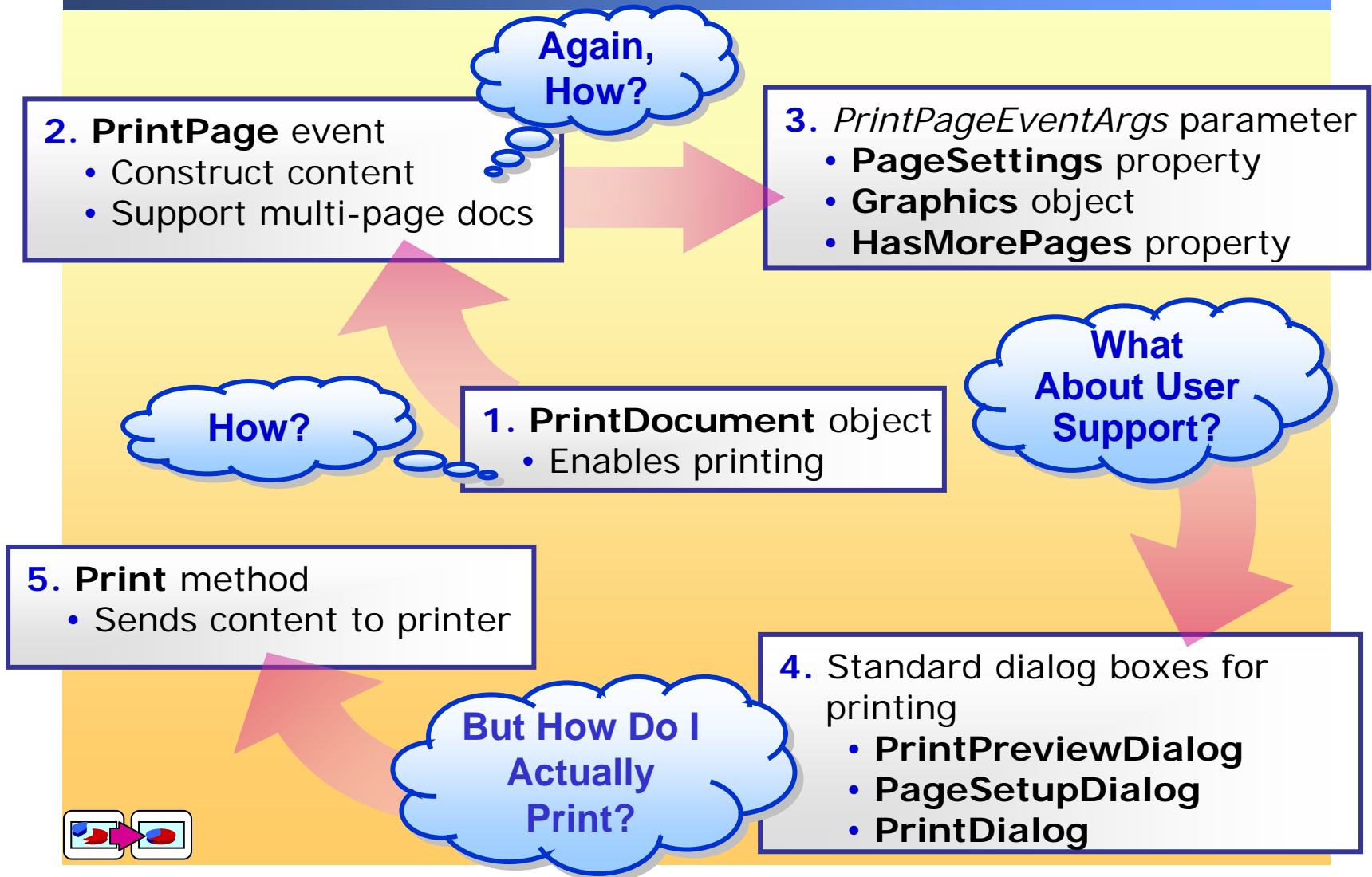
Overview

- Printing from a Windows Forms Application
- Using the Print Preview, Page Setup, and Print Dialogs
- Constructing Print Document Content by Using GDI+
- Creating Reports by Using Crystal Reports

Lesson: Printing From a Windows Forms Application

- How Printing Works in a .NET Windows Forms Application
- PrintDocument Object
- PrintPage Event and PrintPageEventArgs
- How to Enable Printing in a Windows Forms Application
- Practice: Adding Print Support to a Windows Forms Application



How Printing Works in a .NET Windows Forms Application




PrintDocument Object

■ PrintDocument object

- Provides the ability to print a document
- Provides properties that describe what to print

PrintDocument Properties	
	DefaultPageSettings
	DocumentName
	PrintController
	PrinterSettings

PrintDocument Events	
	BeginPrint
	EndPrint
	PrintPage
	QueryPageSettings

PrintDocument Methods	
	Dispose
	Print

Note: The check marks indicate the most commonly used properties, events, and methods of the PrintDocument object

PrintPage Event and PrintPageEventArgs

PrintPage Event

- The **PrintPage** event includes two parameters: an object representing the Sender and a

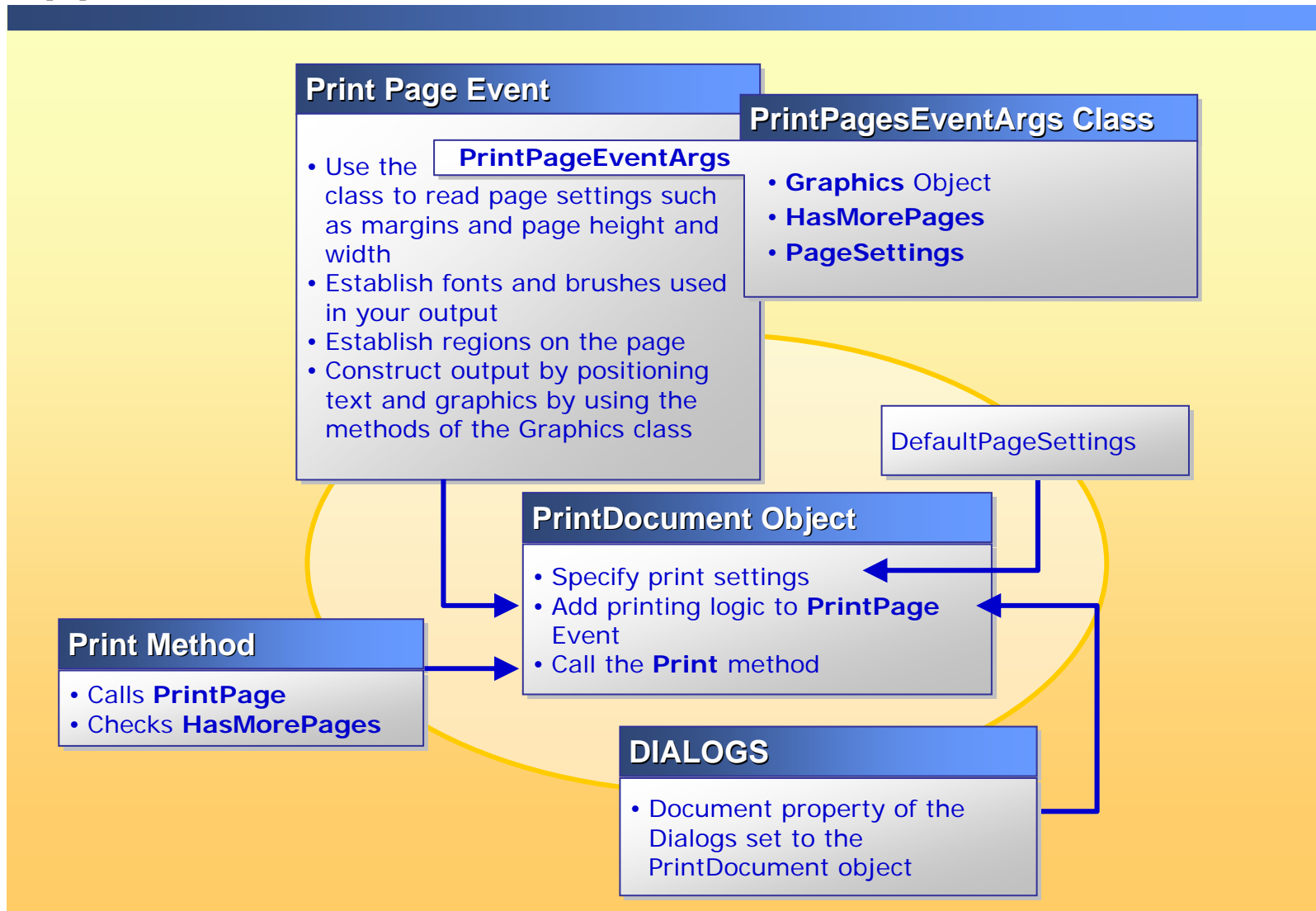
PrintPageEventArgs object

- The **PrintPage** event procedure contains the programming logic that constructs the contents of the print document and ensures that the entire print document is printed

PrintPageEventArgs Object

- The **PrintPageEventArgs** object provides the components required to construct the pages of the print document
- **PageSettings**, **Graphics**, and **HasMorePages** are the critical members

How to Enable Printing in a Windows Forms Application



Practice: Adding Print Support to a Windows Forms Application



In this practice, you will

- Add a **PrintDocument** control to a project
- Create a **PrintPage** event procedure and develop code to construct the contents of a print document
- Set the value of the **HasMorePages** property to indicate when more pages should be printed

Begin reviewing the objectives
for this practice activity



Lesson: Using the Print Preview, Page Setup, and Print Dialogs

- How to Use the PrintPreviewDialog Control
- Practice: Using the PrintPreviewDialog Control
- How to Use the PageSetupDialog Control
- Practice: Using the PageSetupDialog Control
- How to Use the PrintDialog Control
- Practice: Using the PrintDialog Control

How to Use the PrintPreviewDialog Control

PrintPreviewDialog

- Use PrintPreviewDialog to display how a document will appear when printed
- To display a print preview of a document
 1. Create an instance of the PrintPreviewDialog component
 2. Set the Document property to the **PrintDocument** object
 3. Configure the desired display settings for the PrintPreviewDialog and the included **PrintPreviewControl**
 4. Display the dialog by using the **ShowDialog** method

[CodeExample](#)

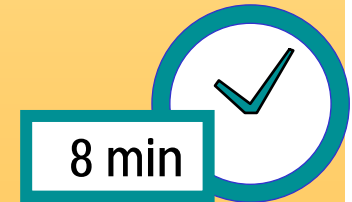
Practice: Using the PrintPreviewDialog Control



In this practice, you will

- Add a **PrintPreviewDialog** control
- Create a **PrintPreview** subroutine
- Modify the display settings for the **PrintPreviewDialog** dialog box

Begin reviewing the objectives for
this practice activity



How to Use the PageSetupDialog Control

PageSetupDialog

- Use PageSetupDialog to set page details in Windows Forms applications
 - Border and margin adjustments
 - Headers and footers
 - Portrait vs. landscape
- To use PageSetupDialog to specify page settings
 1. Create an instance of the PageSetupDialog control
 2. Use ShowDialog to display the dialog at run time
 3. Specify document settings by using the DefaultPageSettings property or PageSettings class

[CodeExample](#)

Practice: Using the PageSetupDialog Control



In this practice, you will

- Add a **PageSetupDialog** control to your application
- Create a **PageSetup** subroutine that displays page settings
- Assign the page settings to the **PrintDocument** object

Begin reviewing the objectives for
this practice activity



How to Use the PrintDialog Control

PrintDialog

- Use PrintDialog to specify printer-related settings
 - Specify printer and print job properties
 - Specify print range and collate settings

- To use a PrintDialog control for users to select a printer
 1. Create an instance of the PrintDialog component
 2. Set the **Document** property to the **PrintDocument** object
 3. Capture user input by using the **DialogResult** property and display the Print dialog box

[CodeExample](#)

Practice: Using the PrintDialog Control



In this practice, you will

- Add a **PrintDialog** control to your application
- Create a **PrintDoc** subroutine that displays the print dialog
- Print a document by using the **PrintDocument.Print** method

Begin reviewing the objectives for
this practice activity

8 min



Lesson: Constructing Print Document Content by Using GDI+

- What Is GDI+?
- What Is the Graphics Object?
- How to Create and Use Pens, Brushes, and Fonts
- How to Draw and Measure Text in the PrintPage Event Procedure
- How to Generate Print Content by Using StreamReader
- Demonstration: Constructing Print Document Content by Using GDI+
- Practice: Constructing Print Document Content by Using GDI+

What Is GDI+?

- **GDI+ (the new graphics device interface)**
 - Enables applications to generate graphics and formatted text for the video display and the printer
 - Allows application developers to create device-independent applications
- **Three parts of GDI+**
 - 2-D vector graphics
 - Text
 - Images

What Is the Graphics Object?

■ Graphics Object:

- Provides the drawing surface on which content is placed
- Provides methods for drawing text and graphics at specified locations
- Provides various tools for modifying its contents

```
Dim myGraphic as Graphics
```

```
' draw lines or outlined shapes using a Pen  
myGraphic.DrawLine(myPen,X1,Y1,X2,Y2)
```

```
' draw filled shapes using a Brush  
myGraphic.FillRectangle(myBrush,X1,Y1,X2,Y2)
```

```
' draw text using a Font and a Brush  
myGraphic.DrawString(myText,myFont,myBrush,X1,Y1)
```

How to Create and Use Pens, Brushes, and Fonts

Pens

A pen is required to draw lines and outlined shapes

```
Dim myPen As New Pen(Color.Blue)
```

Brushes

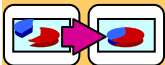
A brush is required to draw filled shapes or draw text

```
Dim myBrush As New SolidBrush(Color.Blue)
```

Fonts

A font is required to draw text of a single size or style

```
Dim myFont As Font("Arial", 16, FontStyle.Bold)
```



How to Draw and Measure Text in the PrintPage Event Procedure

- To draw text

1. Calculate the location for the text
2. Select the Font and Brush that you want to use for this text
3. Call the **Graphics.DrawString** method

```
e.Graphics.DrawString(myText, myFont, myBrush, X1, Y1)
```

- To measure text

```
textWidth = e.Graphics.MeasureString(myText,  
    myFont).Width  
textHeight = e.Graphics.MeasureString(myText,  
    myFont).Height
```

How to Generate Print Content by Using StreamReader

Use StreamReader to read lines of information from a standard text file

```
Private Sub btnPrintPreviewControl_Click(ByVal sender As System.Object, ByVal e As _  
System.EventArgs) Handles _  
btnPrintPreviewControl.Click  
    Try  
        StreamToPrint = New  
StreamReader("PrintMe.Txt")  
        Try  
            PrintPreviewControl()  
  
        Finally  
            StreamToPrint.Close()  
  
    End Try
```

Demonstration: Constructing Print Document Content by Using GDI+



In this demonstration, you see how to construct print document content by using GDI+

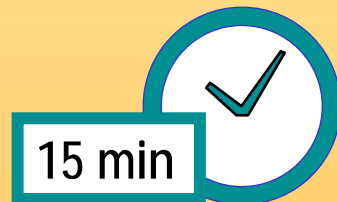
Practice: Constructing Print Document Content by Using GDI+



In this practice, you will

- Create pens, brushes, and fonts
- Create 2-D vector objects
- Measure and position text on the page

Begin reviewing the objectives
for this practice activity



Lesson: Creating Reports by Using Crystal Reports

- Crystal Reports
- How to Create and Format a Report by Using Crystal Reports
- How to View a Report by Using Crystal Report Viewer
- How to Add DataSets to a Report
- Practice: Creating and Viewing Crystal Reports

Crystal Reports

Crystal Reports

- Is the standard reporting tool in .NET
- Allows you to create a report from the beginning or use one of the Report Expert Wizards

Benefits

- You can use any programming language
- Report viewers for Windows-based and Web applications
- Run-time customization of reports
- Easy interaction with reports
- Data visualization and analysis capabilities

<http://msdn.microsoft.com/vstudio/partners/tools/crystaldecisions.asp>

How to Create and Format a Report by Using Crystal Reports



How to View a Report by Using Crystal Report Viewer

To View a Report by Using Crystal Report Viewer

- 1** Add the **Crystal Report Viewer** control to the form
- 2** Set **ReportSource** to the name of the report that is displayed
- 3** Use **ShowZoomButton** to zoom in and zoom out on the report
- 4** Use **ShowPrintButton** to print the report

How to Add DataSets to a Report

- Adding datasets to reports allows you to create reports that are disconnected from the database
- To add datasets to a report, add the code to the Form_Load event

```
Private rpt As New OrderHistory()  
Private dsReportInfo As New DataSet()  
dsReportInfo.ReadXml("NorthwindData.xml")  
rpt.SetDataSource(dsReportInfo)  
CrystalReportViewer1.ReportSource = rpt
```

Practice: Creating and Viewing Crystal Reports



In this practice, you will

- Configure the **SqlConnection1** control on Form1 to connect to the **pubs** database
- Create a Crystal Report
- Add a **Crystal Report Viewer** control to Form1

Begin reviewing the objectives for
this practice activity

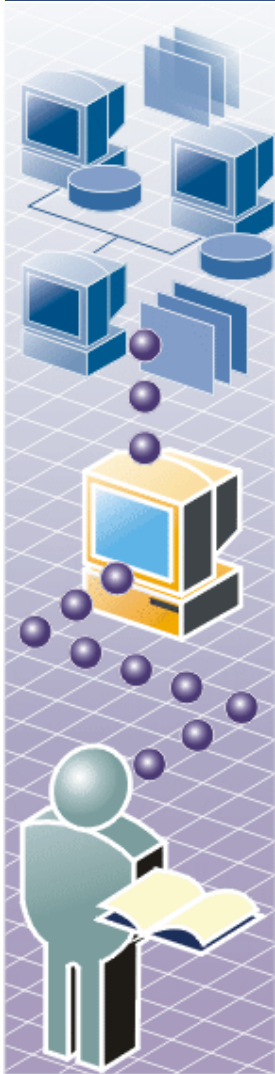
12 min



Review

- Printing from a Windows Forms Application
- Using the Print Preview, Page Setup, and Print Dialogs
- Constructing Print Document Content by Using GDI+
- Creating Reports by Using Crystal Reports

Lab 6.1: Printing Formatted Documents



- Exercise 1: Adding Print Support to an Application
- Exercise 2: Creating Printed Output by Using GDI+